

# Fotografía Computacional: modelo proyectivo de formación de imágenes

Sergio Robles Amieva

TRABAJO DE FIN DE GRADO EN FÍSICA



UNIVERSIDAD COMPLUTENSE DE MADRID

Doble Grado en Matemáticas y Física

Junio 2019

Director:

Juan Antonio Quiroga Mellado

## **Resumen**

La fotografía computacional y en particular el proceso de formación de imágenes se construyen sobre la base del modelo proyectivo. En consecuencia, el estudio del mismo es el punto de partida para entender los procesos y las transformaciones que se llevan a cabo entre los objetos del mundo real y sus representaciones en las imágenes. A lo largo de esta memoria se desarrollará el modelo proyectivo -previo paso por la teoría matemática subyacente, la geometría proyectiva- a fin de presentar un marco consistente que explique las transformaciones que tienen lugar en los procesos de formación de imagen. Además, como culminación de este desarrollo se llevará a cabo la calibración de una cámara de teléfono móvil, que permitirá materializar los conceptos introducidos en el desarrollo teórico presentando una aplicación real y tangible del modelo proyectivo.

## **Abstract**

Computational photography and specially image formation process are built based on the projective model theory. Thus, the comprehension of this model is the starting point in order to understand the processes and transformations that occur in between real world objects and their image representations. All throughout this report the projective model will be developed -after an introduction of the underlying mathematical theory, projective geometry- aiming to present a consistent analytical framework to the transformations taking place in the image formation process. Moreover, a completion to this development will be offered by carrying out a calibration of a smartphone camera, along which all the concepts before explained will materialize, so a real and practical application of the projective model can be presented.

# Índice general

<b>Tabla de contenidos</b>	<b>I</b>
<b>1. Introducción</b>	<b>1</b>
1.1. La fotografía computacional y el modelo proyectivo . . . . .	1
1.2. Objetivos . . . . .	2
1.3. Estructura de la memoria . . . . .	2
<b>2. Fundamento matemático</b>	<b>3</b>
2.1. Introducción a la geometría proyectiva . . . . .	3
2.2. El espacio proyectivo . . . . .	4
2.3. Aplicaciones proyectivas . . . . .	5
2.4. Las coordenadas homogéneas . . . . .	6
2.5. Relación entre el espacio afín y el espacio proyectivo . . . . .	7
<b>3. Modelo proyectivo de formación de imágenes</b>	<b>8</b>
3.1. El modelo <i>pinhole</i> . . . . .	8
3.2. Punto principal, rotación y traslación . . . . .	10
3.3. El modelo de cámara CCD y el skew . . . . .	11
3.4. Correcciones al modelo <i>pinhole</i> . . . . .	12
3.5. Una aplicación inmediata del modelo proyectivo . . . . .	13
3.5.1. La homografía entre dos planos . . . . .	13
3.5.2. La determinación de la homografía . . . . .	14
<b>4. Calibración de una cámara de teléfono móvil</b>	<b>17</b>
4.1. La <i>toolbox</i> . . . . .	17
4.2. Procedimiento . . . . .	18
4.2.1. Inicialización . . . . .	18

4.2.2.	La extracción de esquinas . . . . .	18
4.2.3.	Fase principal de calibración . . . . .	19
4.3.	Resultados . . . . .	20
4.3.1.	Los parámetros intrínsecos . . . . .	20
4.3.2.	El modelo de distorsión . . . . .	21
4.3.3.	Los parámetros extrínsecos . . . . .	22
4.3.4.	Evolución del error en el cálculo de la posición de las esquinas . . . .	22
4.4.	Conclusiones . . . . .	24
<b>Bibliografía</b>		<b>25</b>
<b>A. Imágenes utilizadas</b>		<b>26</b>
<b>B. Resultados de la calibración</b>		<b>27</b>
<b>C. Código del programa de determinación de la homografía</b>		<b>28</b>

# Capítulo 1

## Introducción

### 1.1. La fotografía computacional y el modelo proyectivo

La fotografía computacional es el campo que estudia la captura y procesamiento de imágenes mediante técnicas digitales, por contraposición a la fotografía clásica, que se sirve de elementos ópticos. La fotografía computacional desempeña un papel clave en el área de la visión computacional, una de las disciplinas con más repercusión y actividad en la actualidad debido a sus numerosas aplicaciones en mundos tan dispares como la automoción (conducción automática de vehículos) o la medicina (algoritmos de diagnóstico de imagen).

Dentro de la fotografía computacional podemos encontrar diferentes modelos, que tienen por objetivo estudiar las relaciones entre los objetos del mundo real con respecto a sus representaciones en las imágenes capturadas. Así, existen diferentes modelos según el proceso de formación de imagen seguido por la cámara, el número de imágenes capturadas, el número de cámaras registrando imágenes o incluso según el movimiento de la cámara a lo largo del tiempo.

Uno de los modelos más sencillos es el modelo proyectivo, que entiende el proceso de formación de imágenes como una transformación de perspectiva. Este modelo sirve como base y punto de partida para la elaboración de modelos más complejos, por lo que juega un papel fundamental dentro del campo de la fotografía computacional, y será por tanto el modelo estudiado en esta memoria.

## 1.2. Objetivos

Esta memoria tiene como objetivo, en primer lugar, presentar de manera clara y comprensible el modelo proyectivo de formación de imágenes. Si bien es imposible incluir un desarrollo exhaustivo en esta memoria, se presentarán todos los elementos clave para conseguir presentar de manera consistente *el esqueleto* del modelo proyectivo.

En consecuencia, será necesario introducir previamente el fundamento matemático del modelo proyectivo: la geometría proyectiva. Se presentarán de manera concisa las nociones vertebradoras de la teoría proyectiva, y se explicarán los conceptos que juegan un papel más importante en el modelo proyectivo.

Por otro lado, se ha tomado como segundo objetivo mostrar el proceso de calibración de una cámara de teléfono móvil. Esta calibración toma como referencia el modelo proyectivo de formación de imágenes, por lo que servirá para visualizar y materializar los conceptos presentados a lo largo de la memoria, y para comprobar algunas de las hipótesis habitualmente utilizadas en el modelo proyectivo.

## 1.3. Estructura de la memoria

En el capítulo 2 se presentará la geometría proyectiva. La sección 2.1 introducirá y explicará el enfoque elegido. En las secciones 2.2, 2.3 y 2.4 se realizará una presentación de algunos de los elementos centrales de la teoría. Finalmente, en la sección 2.5 se analizará un concepto clave para el estudio del modelo proyectivo.

En el capítulo 3 se realizará un desarrollo del modelo proyectivo. En la sección 3.1 se presentará el enfoque básico. En las secciones 3.2 y 3.3 se generalizará este modelo, y en la sección 3.4 se darán cuenta de las correcciones necesarias. Finalmente, en la sección 3.5 se presentará brevemente una aplicación del modelo proyectivo.

En el capítulo 4 se presentará la calibración de la cámara de un teléfono móvil. En la sección 4.1 se introducirá la herramienta utilizada. En la sección 4.2 se explicará el procedimiento, cuyos resultados serán analizados en la sección 4.3. Por último, en la sección 4.4 se reflejarán las conclusiones obtenidas.

# Capítulo 2

## Fundamento matemático

### 2.1. Introducción a la geometría proyectiva

De acuerdo con José M. Rodríguez-Sanjurjo y Jesús M. Ruiz, “el origen de la Geometría Proyectiva está asociado a los problemas de perspectiva que aparecen en el Renacimiento al tratar de representar en un plano los objetos del espacio. Este tipo de necesidades, surgidas en la pintura, la arquitectura y la técnica, llevaron a la creación de una teoría matemática que recogiera las bases geométricas subyacentes a los métodos concretos que poco a poco se fueron elaborando en esos campos.” [1].

En la presente sección se realizará una introducción a la geometría proyectiva y sus nociones principales. Si bien no es posible ni razonable hacer en esta memoria un desarrollo exhaustivo de la geometría proyectiva, es conveniente introducir formalmente tanto los espacios donde se trabaja como los elementos y herramientas principales que serán usados durante el desarrollo del modelo proyectivo de formación de imágenes (capítulo 3).

Para ello, tomaremos como punto de partida el álgebra lineal. En particular, serán usadas las nociones de espacio vectorial y geometría euclídea, y en menor medida conceptos de geometría afín. Si bien en la literatura de la visión computacional es habitual introducir la geometría proyectiva partiendo de elementos proyectivos en el plano e ir generalizando los conceptos[2], se ha escogido para esta memoria el enfoque analítico usado generalmente en matemáticas[1] dado que -aunque menos explicativo- es más conciso y útil para presentar los elementos que nos interesan.

## 2.2. El espacio proyectivo

Comencemos introduciendo los espacios donde se trabaja en geometría proyectiva. Sea  $V$  un espacio vectorial sobre un cuerpo  $\mathbb{K}$ . Dados dos elementos  $x, y \in V$  consideramos la siguiente relación de equivalencia

$$x \sim y \iff \exists \lambda \in \mathbb{K}, \quad \lambda \neq 0 \quad \text{tal que} \quad x = \lambda y \quad (2.1)$$

Esta relación de equivalencia identifica todos los vectores pertenecientes a la misma recta vectorial. El espacio que surge de considerar cada una de estas rectas vectoriales como un mismo elemento es, precisamente, el espacio proyectivo.

**Definición 2.2.1.** Se llama *espacio proyectivo* asociado a  $V$  y se denota  $\mathbb{P}(V)$  al espacio cociente inducido por la relación de equivalencia (2.1) en  $V \setminus \{0\}$ , esto es,

$$\mathbb{P}(V) := (V \setminus \{0\}) / \sim$$

De la elección del cuerpo  $\mathbb{K}$  sobre el que se construya el espacio vectorial dependerá el tipo de espacio proyectivo resultante. De esta manera, de los espacios vectoriales reales (i.e.  $K = \mathbb{R}$ ) resultarán los espacios proyectivos reales, de los espacios vectoriales complejos surgirán los espacios proyectivos complejos, etc. A lo largo de esta memoria solo haremos uso de los primeros, por de aquí en adelante se dará por hecho que  $\mathbb{K} = \mathbb{R}$  sin hacer mención específica.

Por otro lado, dependiendo de la dimensión del espacio vectorial  $V$  surgirán diferentes espacios proyectivos asociados. En general, el espacio vectorial  $\mathbb{R}^{n+1}$  da lugar a un espacio proyectivo de dimensión  $n$ , que se denotará de la siguiente manera

$$\mathbb{P}^n = \mathbb{P}(\mathbb{R}^{n+1})$$

Reciben además denominación especial los casos  $\mathbb{P}^1$ , que se conoce como recta proyectiva, y  $\mathbb{P}^2$ , que se conoce como plano proyectivo.

En analogía con el papel que juegan los subespacios vectoriales en el álgebra lineal, en la geometría proyectiva contamos con la siguiente noción.



**Definición 2.2.2.** Una *variedad proyectiva* es un subconjunto  $X$  de un espacio proyectivo  $\mathbb{P} = \mathbb{P}(E)$  de la forma  $X = \mathbb{P}(\hat{X})$  para un cierto subespacio vectorial  $\hat{X} \subset E$ .

En otras palabras, una variedad proyectiva será un subconjunto del espacio proyectivo tal que ella misma sea la proyección de un subespacio vectorial del espacio vectorial original.

## 2.3. Aplicaciones proyectivas

Las aplicaciones proyectivas juegan un papel análogo al de las aplicaciones lineales en el álgebra lineal, y de hecho serán definidas a partir de una aplicación lineal.

**Definición 2.3.1.** Dados dos espacios proyectivos  $X = \mathbb{P}(\hat{X})$  e  $Y = \mathbb{P}(\hat{Y})$ , una *aplicación proyectiva* es una aplicación  $h : X \rightarrow Y$  asociada a una aplicación lineal  $\hat{h} : \hat{X} \rightarrow \hat{Y}$  mediante la siguiente relación:<sup>1</sup>

$$h(x) = [\hat{h}(u)] \quad \text{para } x = [u], \quad u \in \hat{X} \setminus \{0\}$$

No obstante, conviene hacer un apunte clave sobre cómo su comportamiento difiere del comportamiento clásico de las aplicaciones lineales. Para vectores tales que  $\hat{h}(u) = 0$ , la aplicación proyectiva no está definida para su clase de equivalencia  $x = [u]$ . Más concretamente, la aplicación proyectiva sólo está definida en la variedad  $Z = \mathbb{P}(\ker \hat{h})$  correspondiente al núcleo de la aplicación lineal subyacente.

Para evitar los problemas surgidos de esta observación se suele trabajar únicamente con aplicaciones proyectivas inyectivas (y por tanto de núcleo vacío). Como es de esperar, una aplicación proyectiva es inyectiva si y sólo si lo es la aplicación lineal mediante la cual está definida. Surge así el tipo de transformación básico de la geometría proyectiva: la homografía.

**Definición 2.3.2.** Una *homografía* es una aplicación proyectiva inyectiva.

---

<sup>1</sup>Los corchetes denotan la clase de equivalencia del elemento bajo la relación (2.1).

## 2.4. Las coordenadas homogéneas

En analogía con las coordenadas cartesianas utilizadas en los espacios vectoriales, en geometría proyectiva nos encontramos con las coordenadas homogéneas. De acuerdo con la definición realizada en la sección 2.2, existe una ambigüedad a la hora de elegir un representante para clase de equivalencia, problema que debemos solventar.

Sea  $P = \mathbb{P}(E)$  un espacio proyectivo de dimensión  $n$ , asociado por tanto a un espacio vectorial  $E$  de dimensión  $n + 1$ . Si  $u = (u_0, \dots, u_n) \in E \setminus \{0\}$  es un vector del espacio vectorial, el elemento de  $P$  correspondiente se denota  $[u] = (u_0 : \dots : u_n)$ . Esta notación hace hincapié en el hecho de que lo importante en elemento del espacio proyectivo es la proporción entre sus coordenadas, en el sentido de que multiplicar todas las coordenadas por el mismo número no nulo no cambia el elemento al que nos estamos refiriendo.

Por supuesto, la elección de las coordenadas proyectivas depende de la base elegida para el espacio vectorial subyacente. Una explicación más detallada de la relación que guardan estos elementos puede encontrarse en el capítulo III del libro "Lecciones de geometría proyectiva" de Rodríguez-Sanjurjo y Ruiz [1]. Para lo que en esta memoria nos compete, bastará con decir que la representación canónica escogida es aquella que normaliza la última coordenada a 1, lo que en el ejemplo del párrafo anterior se traduciría en escoger

$$[u] = \left( \frac{x_0}{x_n} : \frac{x_1}{x_n} : \dots : 1 \right)$$

Esta elección da un papel especial -por definición- a todos aquellos elementos cuya última coordenada sea cero, pues no van a poder ser normalizados a la forma anterior. El conjunto de esos elementos forma un hiperplano conocido como hiperplano del infinito, que juega un papel fundamental en la geometría proyectiva, como veremos en la siguiente sección.

*Observación 2.4.1.* Durante el desarrollo del modelo proyectivo en el capítulo 3 se usarán exclusivamente las coordenadas homogéneas. Por ello, se ha decidido tomar de aquí en adelante la notación habitual de los vectores  $(a, b, c, \dots, 1)$  para representar las coordenadas homogéneas, y aclararlo allí dónde sea necesario.

## 2.5. Relación entre el espacio afín y el espacio proyectivo

La relación el espacio afín y el espacio proyectivo juega un papel clave en la geometría proyectiva, y en especial en la parte que compete al modelo proyectivo de formación de imágenes. Es por ello que, pese a tratarse de un concepto relativamente complejo, se ha tratado de dar una noción general en la presente sección.

La idea general que relaciona estos dos conceptos es que todo espacio afín se puede *sumergir* en un espacio proyectivo, o recíprocamente, todo espacio proyectivo se puede entender como un espacio afín al que se le ha añadido un hiperplano: *el hiperplano del infinito*. Esta observación será usada continuamente en el tratamiento del modelo proyectivo de formación de imágenes (capítulo 3). Veamos cómo se formaliza este concepto.

Sea  $X$  un espacio afín de dimensión  $n$ . Entonces  $X$  se puede sumergir como hiperplano afín no vectorial de un espacio vectorial  $\hat{X}$  de dimensión  $n + 1$  de modo que su traslación al origen es el espacio vectorial de direcciones de  $X$ , denotado como  $\vec{X}$ . De esta manera, tomando cualquier elemento  $a \in X$  se tiene  $X = a + \vec{X} \subset \hat{X}$ .

Considerando el espacio proyectivo  $P = \mathbb{P}(\hat{X})$  podemos identificar el hiperplano afín con un cierto subconjunto de  $P$ , resultante precisamente de proyectar el hiperplano afín.

Además, se puede demostrar que los elementos del espacio proyectivo  $P$  provienen de un elemento del espacio afín  $X$  si y sólo si no son paralelos a  $X$ . El conjunto de los puntos paralelos a  $X$  forma un hiperplano de  $\hat{X}$  y por ende su proyección constituye un hiperplano de  $P$ . Este hiperplano se conoce como hiperplano del infinito.

*Observación 2.5.1.* Una desarrollo formal y más detallado de todos estos conceptos puede encontrarse en el libro "Lecciones de Geometría Proyectiva", capítulo IV [1].

# Capítulo 3

## Modelo proyectivo de formación de imágenes

En la presente sección se presentará un modelo analítico del proceso de formación de imágenes, fundamentado en las nociones y conceptos introducidos en el capítulo 2.

### 3.1. El modelo *pinhole*

En primer lugar, hay que explicar qué función juega el elemento central del modelo proyectivo de formación de imágenes: la cámara. Analíticamente, podemos entender una cámara como una aplicación que transforma puntos del espacio real de 3 dimensiones en puntos de un plano conocido como plano de la imagen. Esta aplicación se trata de una aplicación proyectiva, de manera que podemos entender el estudio de una cámara o de un modelo proyectivo como el estudio de la matriz de la aplicación proyectiva.

Vamos a considerar la proyección central de puntos del espacio en un plano, donde el centro de la proyección  $C$  será el origen de coordenadas. En el modelo *pinhole* el eje de la proyección,  $Z$  en nuestro caso, se conoce como eje principal, el plano  $Z = f$  será conocido como plano imagen o plano focal, y el punto en el eje principal interseca al plano imagen se conoce como punto principal. Así, un punto del espacio con coordenadas  $\mathbf{X} = (X, Y, Z)^T$  es proyectado al punto en el que la línea que une  $\mathbf{X}$  con el centro de proyección interseca al plano imagen, como se puede ver en la figura 3.1.

La expresión analítica de la transformación proyectiva se obtiene por el teorema de Tales[ver 2, cap 6], y teniendo en cuenta que en el plano imagen la última coordenada es constante ( $Z = f$ ) y puede ser omitida, se tiene que la proyección central descrita viene dada por la expresión (3.1).

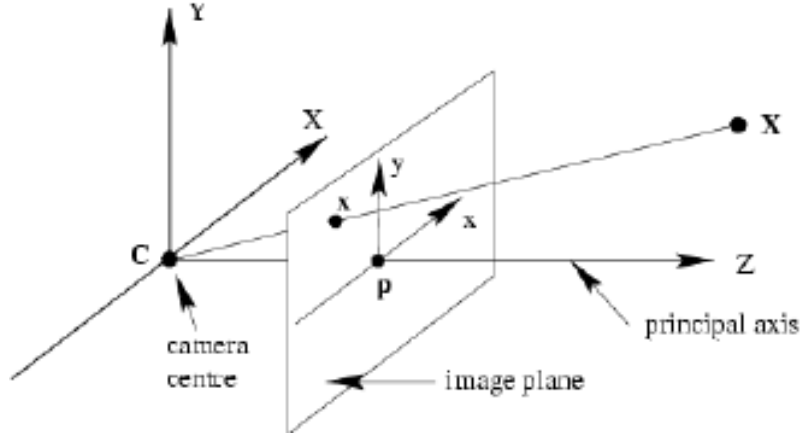


Figura 3.1: Geometría del modelo *pinhole*. En la imagen  $C$  representa el centro de proyección, y  $p$  representa el punto principal, situado en el plano imagen. Origen de la imagen: [3].

$$(X, Y, Z)^T \mapsto \left( f \frac{X}{Z}, f \frac{Y}{Z} \right) \quad (3.1)$$

Entendiendo el mundo real como un espacio afín inmerso en un espacio proyectivo -tal y como se explicó en el apartado 2.5- podemos entender los puntos del mundo real y los puntos de la imagen como vectores homogéneos. La proyección central (3.1) se expresa entonces como una transformación lineal entre las coordenadas homogéneas de los mismos, esto es,

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.2)$$

A la matriz de la expresión (3.2) se le conoce como matriz homogénea de la proyección central, y se le denota  $P$ . Si denotamos además el punto de la imagen como  $\mathbf{x}$ , podemos finalmente escribir la expresión anterior de forma más compacta:

$$\mathbf{x} = P\mathbf{X} \quad (3.3)$$

## 3.2. Punto principal, rotación y traslación

En este apartado vamos a ampliar el modelo presentado en el apartado anterior. Concretamente, vamos a considerar la posibilidad de que los sistemas coordenadas del mundo y de la imagen se encuentren en una posición relativa arbitraria.

En primer lugar, teniendo en cuenta que el origen de coordenadas en la imagen puede no ser el punto principal, deberemos considerar en la aplicación (3.1) que un cierto factor  $p_x$  y  $p_y$  pueda ser sumado a cada coordenada. En coordenadas homogéneas, aprovechando el factor de escala libre, podemos reescribir la transformación (3.2) de la siguiente manera:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{pmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.4)$$

A la matriz de la expresión (3.4) se le conoce como matriz de calibración de la cámara, y se le denota  $K$ .

En segundo lugar, los puntos en el espacio estarán expresados con respecto a un sistema de coordenadas arbitrario, conocido como sistema de coordenadas del mundo. Dos sistemas de coordenadas euclídeos están relacionados por una rotación y una traslación, luego el cambio de las coordenadas del sistema del mundo  $\mathbf{X}$  a las del sistema de coordenadas de la cámara  $\mathbf{X}_c$  será de la forma

$$\mathbf{X}_c = \begin{pmatrix} R & -RC \\ 0 & 1 \end{pmatrix} \mathbf{X}, \quad (3.5)$$

dónde  $R$  es una matriz de rotación  $3 \times 3$  y  $C$  es el vector (no homogéneo) del punto principal en el sistema de coordenadas del mundo.

Suele ser conveniente no hacer explícito el punto principal sino dejarlo implícito en la transformación  $\mathbf{X}_c = R\mathbf{X} + \mathbf{t}$ . Juntando esto con la expresión (3.4) llegamos finalmente a la siguiente matriz de cámara:

$$P = K[R|\mathbf{t}] \quad (3.6)$$

Recapitulando, vemos que en transformación general tenemos 3 grados de libertad en  $K$ , conocidos como parámetros internos, y 6 grados de libertad en  $R$  y  $C$ , conocidos como parámetros externos.

### 3.3. El modelo de cámara CCD y el skew

En fotografía computacional, las imágenes no son un espacio continuo, sino que están compuestas por un conjunto de  $M \times N$  píxeles. Por tanto, será necesario introducir en la matriz de la cámara dos coeficientes  $m_x, m_y$  correspondientes al número de píxeles por unidad de distancia en cada dirección (cabe la posibilidad de que los píxeles tengan forma rectangular: este tipo de cámara se conoce como CCD). De esta manera, la forma de la matriz de calibración de una cámara CCD es:

$$K = \begin{pmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.7)$$

donde  $\alpha_{x,y} = f m_{x,y}$  y  $(x_0, y_0)$  son las coordenadas del punto principal en píxeles, es decir,  $x_0 = m_x p_x$ ,  $y_0 = m_y p_y$ .

Existe además la posibilidad de que en algunos modelos antiguos de cámara los ejes X e Y de la cuadrícula de píxeles no estén completamente alineados. Por ello, es necesario introducir un factor  $s$  de sesgo o *skew*, que junto con la expresión 3.7 queda:

$$K = \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.8)$$

No obstante, este nuevo parámetro intrínseco introducido será prácticamente nulo en las cámaras modernas. En cualquier caso, podemos escribir la transformación más general como

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K[R|t] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.9)$$

### 3.4. Correcciones al modelo *pinhole*

Hay que recordar que el modelo *pinhole* es sólo una aproximación de la proyección que tiene lugar realmente en una cámara. Si bien nos ofrece un modelo útil que establece una relación lineal entre las coordenadas de un objeto en el sistema del mundo y el de la imagen, sus predicciones dejan de ser acertadas cuando se requiere un alto grado de precisión. En estos casos es necesario aplicar una serie de correcciones.

La corrección que juega un papel más importante es aquella que trata la distorsión radial causada por la lente, y que produce que los puntos sean desplazados radialmente en el plano de la imagen [4]. Esta distorsión radial hace que el proceso de formación de imagen deje de ser lineal, y puede ser aproximada mediante la siguiente expresión (ver [5]):

$$\begin{pmatrix} \delta u \\ \delta v \end{pmatrix} = \begin{pmatrix} x(k_1 r_i^2 + k_2 r_i^4 + \dots) \\ y(k_1 r_i^2 + k_2 r_i^4 + \dots) \end{pmatrix} \quad (3.10)$$

donde  $k_1, k_2, \dots$  son los nuevos coeficientes introducidos,  $x, y$  se refieren a las coordenadas en el plano de la imagen tras la transformación de escala (3.2) y  $r = \sqrt{x^2 + y^2}$ .

Por otro lado, los centros de curvatura de la superficie de las lentes no son siempre estrictamente colineales, de donde surge otro tipo de distorsión bastante común conocida como distorsión descentrante[4]. Este tipo de distorsión tiene componentes tanto radial como tangencial, y puede ser modelizada de acuerdo con la siguiente expresión:

$$\begin{pmatrix} \delta u \\ \delta v \end{pmatrix} = \begin{pmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y^2) + 2p_2 x^2 y^2 \end{pmatrix} \quad (3.11)$$

donde  $p_1, p_2$  son los nuevos coeficientes introducidos.

*Observación 3.4.1.* Otros tipos de distorsión han sido estudiados, como la distorsión lineal cuando los ejes de la imagen no son ortogonales [6], o la distorsión de prisma, producida por imperfecciones en las lentes [7]. En esta memoria nos limitaremos el estudio de la distorsión a las componentes radial y descentrante.



## 3.5. Una aplicación inmediata del modelo proyectivo

A continuación, vamos a desarrollar un ejemplo de aplicación del modelo proyectivo. Conocida la posición de 4 puntos coplanares en la imagen, es posible determinar la homografía que describe la transformación entre el plano de los puntos y su representación en la imagen. De esta manera, una vez conocida la homografía nos permitirá medir ángulos o longitudes relativas en el plano de estudio, lo que ofrece numerosas posibilidades. Para el desarrollo introduciremos primero en el apartado 3.5.1 la descripción de la homografía entre dos planos. Después, en el apartado 3.5.2, explicaremos como determinarla a partir de las coordenadas de 4 puntos conocidos.

*Observación 3.5.1.* Una implementación en MATLAB del cálculo de la homografía se ha realizado como ejercicio, y se puede ver en el anexo C.

### 3.5.1. La homografía entre dos planos

Vamos a retomar el desarrollo del modelo proyectivo de la sección 3.2, considerando una transformación de proyección de perspectiva de una lente con distancia focal  $f$  junto con una roto-traslación. Para este desarrollo podemos considerar que el origen de coordenadas de la imagen se encuentra en el punto principal, y omitiremos por tanto los parámetros  $p_x$  y  $p_y$ .

Sean  $(X, Y, Z)^T$  las coordenadas de un objeto en el sistema de coordenadas externo, y sean  $(X, Y, Z, 1)^T$  sus coordenadas homogéneas asociadas. Tenemos que la transformación del sistema viene descrita por

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.12)$$

donde  $[u, v, 1]^T$  representan las coordenadas homogéneas en el sistema de la imagen.

Vamos a estudiar cómo puede modelizarse la transformación de un único plano del mundo real al plano de la imagen. Si tomamos el origen de coordenadas externas de tal manera

que nuestro plano de estudio coincida con el plano  $Z = 0$ , podemos imponer esta misma condición en la ecuación (3.12) para simplificar y llegar a

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fr_{11} & fr_{12} & ft_x \\ fr_{21} & fr_{22} & ft_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

O, interpretando esta última expresión como una homografía correspondiente a la transformación proyectiva plana podemos reescribirlo como

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (3.13)$$

### 3.5.2. La determinación de la homografía

Partimos de la base de que existen cuatro puntos pertenecientes al plano de estudio cuyas coordenadas conocemos en ambos sistemas de referencia, esto es, conocemos cuatro pares  $\{(X_i, Y_i), (u_i, v_i)\}$   $i = 1, 2, 3, 4$  verificando la ecuación 3.13. Podemos reescribir esta ecuación de la siguiente manera

$$\begin{aligned} u &\sim h_{11}x + h_{12}y + h_{13} \\ v &\sim h_{21}x + h_{22}y + h_{23} \\ 1 &\sim h_{31}x + h_{32}y + h_{33} \end{aligned} \implies \begin{cases} u = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \\ v = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \end{cases} \quad (3.14)$$

Nótese que en la ecuación 3.13 no existen realmente 9 grados de libertad dado que se trata de una relación de proporcionalidad y no de igualdad (estamos tratando con coordenadas homogéneas). Como resultado de esto, notamos que en la expresión 3.14 podemos multiplicar todos los  $h_{ij}$  por un mismo factor no nulo sin alterar las igualdades. Podemos aprovechar este grado de libertad *extra* para forzar  $h_{33} = 1$ , de manera que podemos escribir,

$$\begin{cases} u = \frac{h_{11}x+h_{12}y+h_{13}}{h_{31}x+h_{32}y+1} \\ v = \frac{h_{21}x+h_{22}y+h_{23}}{h_{31}x+h_{32}y+1} \end{cases} \implies \begin{cases} u + uh_{31}x + uh_{32}y = h_{11}x + h_{12}y + h_{13} \\ v + vh_{31}x + vh_{32}y = h_{21}x + h_{22}y + h_{23} \end{cases}$$

y resolviendo de manera implícita para  $u$  y  $v$  llegamos a

$$\begin{cases} u = h_{11}x + h_{12}y + h_{13} - uh_{31}x - uh_{32}y \\ v = h_{21}x + h_{22}y + h_{23} - vh_{31}x - vh_{32}y \end{cases} \quad (3.15)$$

Recordemos que esta igualdad se produce para cuatro pares de puntos  $(x_i, y_i), (u_i, v_i)$   $i = 1, 2, 3, 4$ . Teniendo en cuenta que nuestras incógnitas son los  $h_{ij}$  es conveniente reescribir la expresión 3.15 en forma matricial de la siguiente manera:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -u_i x_i & -u_i y_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -v_i x_i & -v_i y_i \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} \quad i = 1, 2, 3, 4 \quad (3.16)$$

Finalmente, podemos yuxtaponer los cuatro casos de  $i = 1, 2, 3, 4$  para dejarlo todo convenientemente escrito en una única ecuación:

$$\begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -v_1x_1 & -v_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -u_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -v_2x_2 & -v_2y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -u_3x_3 & -u_3y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -v_3x_3 & -v_3y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -u_4x_4 & -u_4y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -v_4x_4 & -v_4y_4 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} \quad (3.17)$$

De esta manera, la determinación de la homografía será equivalente al resolver el sistema anterior.

*Observación 3.5.2.* El desarrollo que venimos de efectuar para la determinación de la homografía parte de la base de que conocemos 4 puntos, que tal y como hemos visto son las condiciones mínimas bajo las que podemos determinar los 8 grados de libertad de la homografía. No obstante, es evidente que conocer más puntos en la imagen llevará a una determinación más precisa de la homografía. La idea a seguir es extender la matriz de la ecuación (3.17) con tantos bloques de la forma (3.16) como puntos dispongamos. Nos quedará entonces un sistema sobredeterminado, cuya solución puede ser encontrada mediante la técnica de optimización de mínimos cuadrados con ayuda de la descomposición en valores singulares (*SVD*). Una explicación completa de esta técnica puede ser encontrada en el apéndice 5 del libro “Multiple View Geometry in Computer Vision” [2].

# Capítulo 4

## Calibración de una cámara de teléfono móvil

De acuerdo con Heikkila y Silvén, “la calibración de una cámara en el contexto de la visión computacional se trata del proceso de determinación las características ópticas y geométricas (parámetros intrínsecos) y/o la determinación de la posición y la orientación del sistema de la cámara con respecto a un cierto sistema de coordenadas externo (parámetros extrínsecos)” [5].

En esta sección se presenta el procedimiento de calibración de una cámara de teléfono móvil. El teléfono móvil es un BQ Aquaris X, y la cámara consta de un sensor Sony IMX298, resolución de  $16MP$ , una apertura de  $f/2,0$ , y un factor de  $1,12\mu m/\text{pixel}$  (ver [8]). Mediante el propio *software* del teléfono se ha desactivado el enfoque automático, y se ha fijado el enfoque al infinito para evitar cambios en la focal de la cámara.

### 4.1. La *toolbox*

Para la calibración se ha usado el software *MATLAB* y la *Camera Calibration Toolbox for MATLAB* de Jean-Yves Bouguet[9], que se basa en la identificación de las esquinas de un tablero en una serie de fotografías realizadas en distintas posiciones. Esta *toolbox* nos permitirá medir los parámetros intrínsecos y extrínsecos de la cámara explicados en el capítulo 3, y ofrece la posibilidad de aplicar al modelo de la cámara las correcciones añadidas en la sección 3.4.

En particular, ofrecerá las distancias focales de la cámara en píxeles  $\alpha_x$  y  $\alpha_y$ , la posición del punto principal en píxeles  $(p_x, p_y)$ , y los dos primeros coeficientes de cada tipo de distorsión  $k_1, k_2, p_1, p_2$ . Después de cada optimización también tendremos una estimación de la

incertidumbre de cada uno de estos parámetros.

Así mismo, la *toolbox* ofrece la posibilidad de medir el *skew* y el tercer coeficiente de distorsión radial, correspondiente al sexto orden. Sin embargo, estos parámetros se han fijado a cero: el primero, por saber que se trata de una cámara con píxeles cuadrados, y el segundo, por ser suficiente con los coeficientes de distorsión radial de segundo y cuarto orden, tal y cómo se explicó en el apartado 3.4.

*Observación 4.1.1.* En la *toolbox*, el origen de coordenadas de la imagen se encuentra en la esquina superior izquierda, lo que deberá ser tenido en cuenta tanto en la calibración como en la interpretación de los resultados.

## 4.2. Procedimiento

### 4.2.1. Inicialización

Se inicializa la *toolbox* en el directorio donde están las imágenes de calibración y se ejecuta la rutina “*Image names*” a fin de indicarle al programa el nombre y formato de las imágenes. Las imágenes son cargadas en la memoria.

### 4.2.2. La extracción de esquinas

El procedimiento de extracción de esquinas consiste en la determinación de la situación de las cuatro esquinas en cada imagen con ayuda del programa<sup>1</sup>. Para ello, se ejecuta la rutina *Extract grid corners*, que irá abriendo las imágenes una a una para que vayamos indicando donde están. Las esquinas deben ser seleccionadas siempre en el mismo orden, y se ha adoptado el convenio de empezar por aquella más cerca de la esquina superior izquierda de la pantalla y proseguir en sentido horario. Además, el programa cuenta los cuadrados del tablero de forma automática, y el tamaño de los mismos es el que viene por defecto: 30mm.

---

<sup>1</sup>La rutina cuenta con un procedimiento automático de detección de esquinas en un cierto área en torno al punto donde indica el usuario. El tamaño del área donde buscar puede ser especificado, no obstante, se ha dejado el valor por defecto que sugiere la propia *toolbox*, de  $48 \times 48mm$ .

Por otro lado, después seleccionar las esquinas en una imagen, el programa predice una primera cuadrícula y pregunta al usuario si esta se adapta bien a las esquinas, o si por el contrario es necesario introducir manualmente una primera suposición del factor de distorsión  $k_C$ . En nuestro caso lo hemos necesitado para la imagen 9,  $k_c = 0,4$ , y para la imagen 17,  $k_c = -0,3$ .

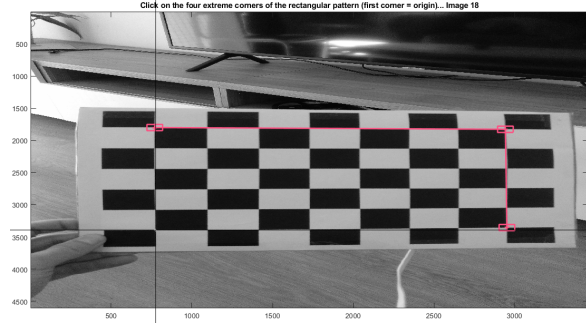


Figura 4.1: Imagen del procedimiento de selección de esquinas.

### 4.2.3. Fase principal de calibración

En esta etapa del procedimiento se lleva a cabo la calibración en dos pasos, un primer paso de inicialización mediante los datos recabados en la fase anterior, y un segundo paso de optimización no lineal por el método del descenso de gradiente. Esta optimización, de tipo mínimos cuadrados para el error de reproyección de las esquinas, se realiza para los 9 parámetros intrínsecos (focal, punto principal, distorsión) y los  $6 * 24$  parámetros extrínsecos (las posiciones y orientaciones de cada fotografía).

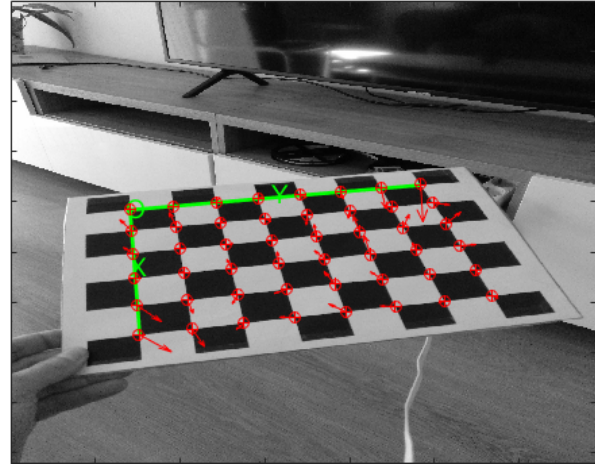


Figura 4.2: Ejemplo de reproyección en la imagen 14.

En nuestra calibración se han requerido 20 iteraciones de descenso de gradiente para llegar a los resultados, que se pueden ver en la tabla B.1 del anexo B. Cabe destacar que en esta fase los parámetros de *skew*  $s$  y el sexto orden de distorsión radial  $k_3$  están fijados a 0, puesto que para la mayoría de casos no son necesarias[9].

Una vez computados los resultados, la función *Reproject on images* nos mostrará las nuevas situaciones de las esquinas en base a los nuevos valores de los parámetros. Un ejemplo gráfico de esta reproyección puede verse en la figura 4.2.

A continuación, podremos recalcular las posiciones de las esquinas en base a los nuevos parámetros mediante la rutina *Recompute corners*. En esta fase el método del gradiente parte de los resultados de la anterior calibración, y suele converger con menos iteraciones.

Una vez realizada esta segunda calibración, podemos concentrarnos en mejorar las proyecciones en las esquinas que más error generen. Para ello, contamos con la herramienta *Analyse error tool*, podremos ver una representación de la estimación del error en cada esquina, y en la que las esquinas de la misma figura vienen representadas por cruces del mismo color. En nuestro caso las imágenes que más error han tenido fueron las imágenes 2, 6, 7, 9 y 17. Una vez identificadas, procederemos a recalcular las esquinas en esas imágenes específicas y realizar una última calibración, que de nuevo partirá de los datos de la última optimización.

## 4.3. Resultados

### 4.3.1. Los parámetros intrínsecos

En primer lugar, vamos a comentar los valores obtenidos para los parámetros objetivo de la calibración con respecto a cada una de las fases de optimización, recogidos en la tabla B.1 del anexo B.

De acuerdo con el desarrollo de la sección anterior, el programa se inicializa a partir de los datos proporcionados por el usuario en la selección de esquinas junto con la detección automática que el mismo realiza. En esta primera fase, todos los parámetros de distorsión, además del *skew*, están fijados a cero: se trata de un modelo lineal, y carecemos todavía de una estimación del error.

En la primera optimización se observan cambios razonablemente significativos en los resultados. Si bien los valores de la focal sólo varían en un 1 %, esta variación supone más



del doble de la incertidumbre en la variable, por lo que podemos afirmar que se ha mejorado la predicción inicial. Para el punto principal los cambios son también del orden del 1 %, y en valor absoluto las variaciones son del orden de la incertidumbre.

Para la segunda optimización, resultante de recalcular las posiciones de las esquinas en base a los valores de la primera optimización, y para la tercera optimización, correspondiente a una corrección en las imágenes que más error arrojaban, se puede afirmar que no se encuentran variaciones significativas en los parámetros. Todos los cambios en los valores son inferiores a un cuarto del valor de las incertidumbres correspondientes. El único nuevo dato que podemos extraer en claro de estas dos últimas optimizaciones es que el primer coeficiente de la distorsión descentrante  $p_1$  se hace claramente compatible con 0; ahora el valor del parámetro es un 20 % inferior al de su incertidumbre.

#### **4.3.2. El modelo de distorsión**

A continuación vamos a comentar el modelo de distorsión obtenido para la cámara. La *toolbox* nos ofrece tanto los valores de los coeficientes de distorsión (tabla B.1 como una representación gráfica del modelo.

En cuanto a los factores de distorsión, observamos que los parámetros correspondientes al segundo y cuarto orden de distorsión radial ( $k_1$  y  $k_2$ , respectivamente) obtienen valores significativos, y muy superiores a  $p_1$  y  $p_2$ .

Por el contrario, los coeficientes correspondientes a la distorsión descentrante tienen valores muy inferiores, y de hecho próximos a cero: para  $p_2$  la incertidumbre es 3 veces mayor que su valor absoluto, por lo que no podemos inferir que se haya encontrado un valor significativo. Para  $p_1$ , ha sido necesaria la segunda optimización para esclarecer que es compatible con cero, pues en la primera el valor de la variable y su incertidumbre eran muy similares.

Una representación del modelo obtenido puede ser observada en la figura 4.3. En consonancia con las magnitudes de cada componente, puede observarse como el modelo de distorsión completa (imagen 4.3a) es prácticamente idéntico al de la distorsión radial por separado (imagen 4.3b).

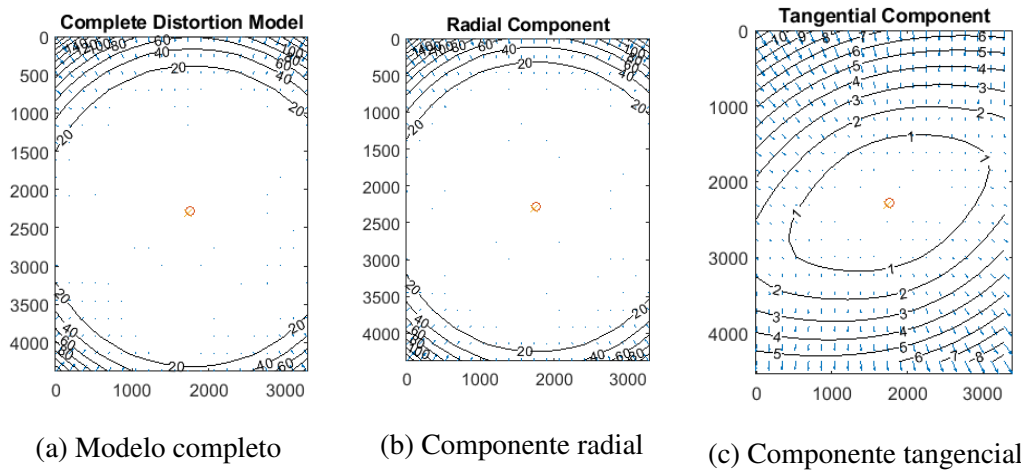


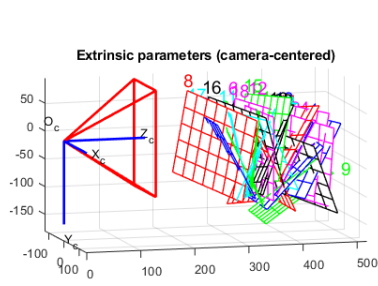
Figura 4.3: Representación de cada uno de los modelos de distorsión, en píxeles. Las flechas indican la dirección y magnitud de la distorsión en cada punto, y los trazos negros representan líneas de nivel. En rojo, el punto principal.

### 4.3.3. Los parámetros extrínsecos

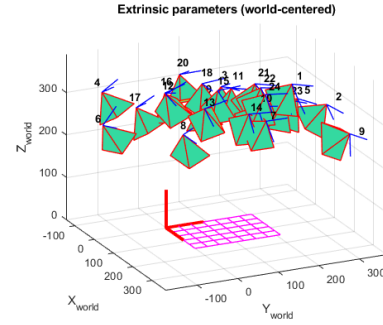
Durante el proceso de calibración de la cámara también se calculan los parámetros extrínsecos correspondientes a cada imagen, esto es, la posición relativa entre la cámara y el tablero de calibración. El propio programa ofrece la opción de visualizarlos de dos maneras: una en la que se considera la posición de la cámara fija y el tablero va variando su posición, y otra en el que se considera el tablero fijo y es la cámara la que se va moviendo. Podemos ver ambas reconstrucciones en la figura 4.4. Comprobamos que no existe ninguna imagen que desentone y que todas las posiciones son razonables: todos los tableros se encuentran a una distancia de entre  $300\text{mm}$  y  $400\text{mm}$  de la cámara en la figura 4.4a, en consonancia con la longitud de el brazo que sujeta el tablero en las fotos.

### 4.3.4. Evolución del error en el cálculo de la posición de las esquinas

Por último, tal y cómo se explico en la sección 4.2 la *toolbox* ofrece una herramienta para analizar el error entre la predicción de la posición de las esquinas y su posición real. En la figura 4.5 se muestra el error en la posición de las esquinas tras cada una de las tres



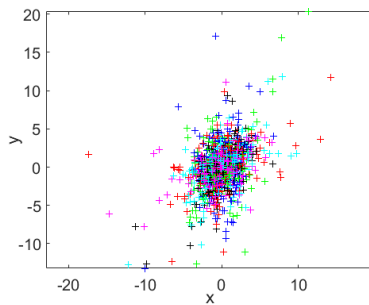
(a) Cámara fija. A la izquierda, el sistema de referencia de la cámara. A la derecha, la posición del tablero en cada foto.



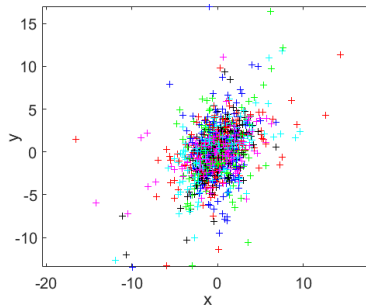
(b) Tablero de calibración fijo. Debajo, el sistema de referencia del tablero. Arriba, la posición de la cámara en cada foto.

Figura 4.4: Reconstrucción gráfica de las posiciones relativas en milímetros entre la cámara y el tablero a partir de los parámetros extrínsecos encontrados.

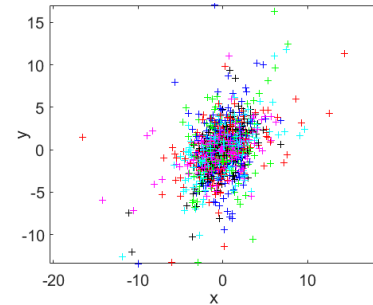
optimizaciones. Podemos observar que de la primera calibración (imagen 4.5a) a la segunda calibración (imagen 4.5b) se produce un descenso general del error de reproyección (nótese el cambio de escala en los ejes). Por el contrario, entre la segunda y tercera calibración (imagen 4.5c) no se ha mejorado significativamente el modelo, pues la distribución de los errores es prácticamente idéntica.



(a) Primera calibración



(b) Segunda calibración



(c) Tercera calibración

Figura 4.5: Representación de los errores de reproyección en píxeles de cada esquina. Un mismo color hace referencia a esquinas de la misma imagen.

## 4.4. Conclusiones

En primer lugar, debemos observar que la primera optimización ha producido cambios significativos en los valores de los parámetros intrínsecos, tal y cómo se analizó en el apartado 4.3.1. Por tanto, podemos concluir que la introducción de correcciones no lineales al modelo *pinhole* (sección 3.4) está justificada.

En segundo lugar, si bien la segunda optimización no ha producido cambios significativos en los valores de las variables, sí que ha contribuido a reducir el error de reproyección sensiblemente, de acuerdo con los resultados expuestos en el apartado 4.3.4. Además, ha ayudado a esclarecer que el primer coeficiente de la distorsión descentrante es compatible con 0, tal y como se explicó al final del apartado 4.3.1. Por tanto, podemos concluir que si bien esta segunda optimización no es imprescindible, ayudar a mejorar el modelo. Por el contrario, la tercera optimización a resultado irrelevante, por lo que -en los márgenes y rangos de la calibración realizada- carece de utilidad.

En tercer lugar, observamos que los resultados descritos en el apartado 4.3.2 están en consonancia con el modelo de distorsión introducido en 3.4, y en particular confirman el hecho de que la distorsión radial juega un papel mucho más importante que la distorsión descentrante.

Por añadidura, podemos utilizar el factor de milímetros por píxel de la cámara para valorar el resultado para la focal. Ya convertido, su valor ha sido de entorno a 4,05 milímetros en ambas direcciones, lo que supone un valor razonable para la cámara de un teléfono móvil.

Como conclusión general, podemos afirmar que el modelo *pinhole* no basta por sí sólo para explicar de forma precisa el funcionamiento de la cámara, y que si bien la distorsión descentrante no tiene un papel importante en la misma, la distorsión radial ha de ser tomada en cuenta para una correcta descripción del proceso de formación de imágenes.

# Bibliografía

- [1] José M. Rodríguez-Sanjurjo y Jesús M. Ruiz. *Lecciones de Geometría Proyectiva*. Sanz y Torres, 2009. ISBN 978-84-96808-37-9.
- [2] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. ISBN 0521540518.
- [3] Pascal Monasse, Jean-Michel Morel, and Zhongwei Tang. Epipolar rectification. 05 2019.
- [4] Alan D. Jones. Manual of photogrammetry. *Cartography*, 12, 09 1982. doi: 10.1080/00690805.1982.10438226.
- [5] Janne Heikkila and Olli Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 1997. URL <http://dl.acm.org/citation.cfm?id=794189.794489>.
- [6] T. Melen. *An examination of keystroke dynamics for continuous user authentication*. PhD thesis, Norges Tekniske Hoegskole, Trondheim, 1994.
- [7] Juyang et. al. Weng. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10):965–980, 1992.
- [8] URL <https://www.bq.com/es/aquaris-x>.
- [9] Jean-Yves Bouguet. Camera calibration toolbox for MATLAB, 2015. URL [www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/).

# Apéndice A

## Imágenes utilizadas

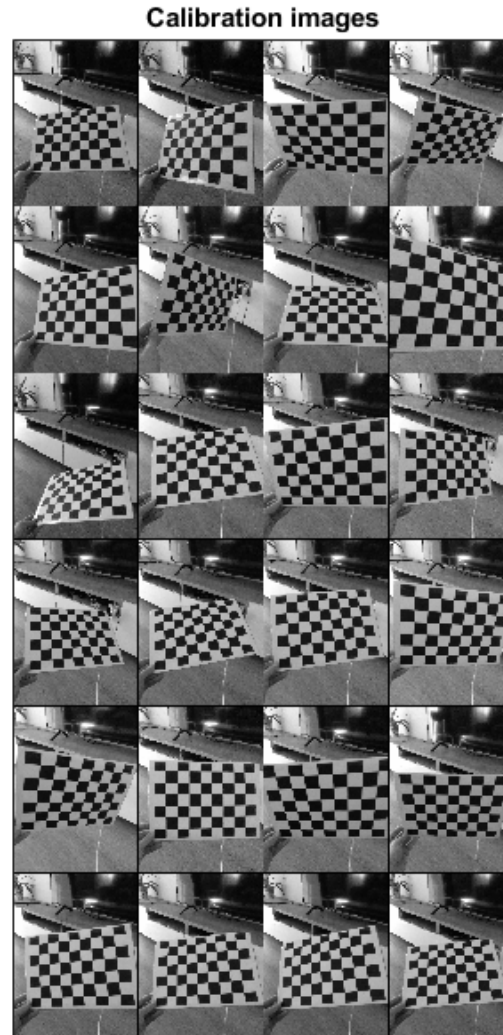


Figura A.1: Las 24 imágenes que han sido usadas para la calibración.

# Apéndice B

## Resultados de la calibración

	Inicialización	Optimización 1		Optimización 2		Optimización 3	
	Valor inicial	Valor	Error	Valor	Error	Valor	Error
$\alpha_x$ (px)	3650	3614	18	3613	17	3613	18
$\alpha_Y$ (px)	3650	3611	18	3611	18	3611	18
$x_0$ (px)	1727	1755	25	1756	25	1755	25
$y_0$ (px)	2303	2287	19	2285	19	2285	19
$k_1$	0	0,044	0,020	0,044	0,019	0,044	0,019
$k_2$	0	-0,255	0,077	-0,257	0,075	-0,257	0,075
$p_1$	0	0,0020	0,0021	0,00161	0,0021	0,00158	0,0021
$p_2$	0	0,0008	0,0023	0,0008	0,0023	0,00073	0,0023

Tabla B.1: Tabla con los parámetros intrínsecos y coeficientes de distorsión obtenidos junto con sus incertidumbres en cada una de las fases de optimización de la calibración.

# Apéndice C

## Código del programa de determinación de la homografía

```
1 %      Esquema de la estructura del algoritmo:
2 %          1. Rutina para la configuracion de los datos:
3 %              1.1 Pregunta si quieres usar los valores por
                defecto , si no:
4 %              1.2 Abre imagen , selecciona puntos [u_i,v_i]
                para i=1,2,3,4.
5 %              1.3 Transformamos los u,v a milímetros. (1.12
                m /pixel)
6 %          2. Rutina que cree la matriz A a partir de {[u_i,
                v_i]} y {[X_i,Y_i]}
7 %          3. Creamos el vector u, resolvemos el sistema  $h=A \backslash h$ 
                y obtenemos la matriz de la homografia.
8
9 %%      RUTINA GLOBAL %%
10 %% 1. Configuracion de los datos
11
12 archivo=selecciona_archivo();
13 %% 2. Crea la matriz
14 A=crea_matriz(u)
15 %% 3. Crea vector vertical u y resuelve sistema. Calcula la
                matriz de la homografia.
16 u_vertical=zeros(8,1);
```



```

17 for i=1:4
18     u_vertical(i*2)=u(i,2);
19     u_vertical(i*2-1)=u(i,1);
20 end
21 h_vertical=A\u_vertical;
22 h=[h_vertical(1),h_vertical(2),h_vertical(3);h_vertical(4),
    h_vertical(5),h_vertical(6);h_vertical(7),h_vertical(8),1]
23
24 %% SUBROUTINAS
25 %% I. selecciona_archivo()
26 % Selecciona archivo o utiliza uno por defecto. Si se
    selecciona un archivo
27 % llama a la funci n selecciona_4puntos, si no, utiliza los
    valores por defecto.
28
29 function archivo=selecciona_archivo()
30     archivo=input("Escribe el nombre de la foto (dejar en
        blanco toma imagen y puntos por defecto): ", 's');
31     if isempty(archivo)
32         archivo='Imagen_Medir.jpg';
33         u=1.0e
            +3*[0.9215,1.7375;1.6835,1.7675;0.6935,2.7335;1.6115,2.7755];
34     else
35         img=imread(archivo);
36         u=selecciona_4puntos(img);
37     end
38

```

```

39     u=1.12e-03*u;    % Transformamos de pixeles a mil metros
40 end
41
42 %% II.  selecciona_4puntos(img)
43 % Selecciona 4 puntos de la imagen. Hay que pulsar 4 veces y
    luego hacer doble click en cualquier lado.
44
45 function u=selecciona_4puntos(img)
46     close all
47     imshow(img)
48     fig=gcf;
49     title("Haz doble click para salir");
50     xlabel('Selecciona cuatro puntos en sentido horario
    empezando por el de "arriba a la izquierda" ');
51     [ui,vi]=getpts(fig);
52     for i=1:4
53         u(i,:)=[ui(i),vi(i)];
54     end
55     close all
56 end
57
58 %% III.  crea_matriz(u)
59 % Crea la matriz A con los valores x e y predefinidos a
    partir de los valores de u
60
61 function A=crea_matriz(u)
62 x=[0,150,150,0];          % Puntos por defecto
63 y=[0,0,210,210];          % Puntos por defecto

```

```

64 Ai=zeros(2,8,4);           % Creamos los cuatro bloques
65
66     for i=1:4
67         Ai(1,1:3,i)=[x(i),y(i),1];
68         Ai(2,4:6,i)=[x(i),y(i),1];
69         Ai(1:2,7:8,i)=[-u(i,1)*x(i),-u(i,1)*y(i); -u(i,2)*x(i)
            ,u(i,2)*y(i)];
70     end
71
72 A=cat(1,Ai(:,:,1),Ai(:,:,2),Ai(:,:,3),Ai(:,:,4)); %
    Yuxtaponemos los bloques
73
74 end

```