

# **Detección de líneas para la estimación y corrección de la distorsión de lentes en imágenes**

Daniel Elías Santana Cedrés

Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería  
Universidad de Las Palmas de G.C.

IUSIANI. Universidad de Las Palmas de G.C.

# Trabajo Fin de Máster

**Título:** Detección de líneas para la estimación y corrección de la distorsión de lentes en imágenes

**Apellidos y nombre del alumno:** Santana Cedrés, Daniel Elías

## Tutores:

**Dr. Luis Álvarez León**

*Catedrático de la Universidad de Las Palmas de Gran Canaria.*

*Área de Conocimiento: Ciencias de la Computación e Inteligencia Artificial.*

**Dr. Javier Lorenzo Navarro**

*Profesor de la Universidad de Las Palmas de Gran Canaria.*

*Área de Conocimiento: Ciencias de la Computación e Inteligencia Artificial.*

IUSIANI. Universidad de Las Palmas de G.C.

# Agradecimientos

Todo trabajo no es sólo fruto personal, si no también de aquellos que nos apoyan, nos ayudan, nos guían o simplemente están ahí. Nunca he sido partidario de hacer simples listas de personas cuando doy las gracias, así que trataré de hacerlo de forma breve.

En primer lugar quiero darle las gracias a mi familia. Sin su apoyo y su paciencia esos días en los que vives más en el trabajo que en casa, las cosas no serían posibles.

Sería inexcusable obviar el agradecimiento a mis tutores: Luis Álvarez y Javier Lorenzo. Sin sus conocimientos y su ayuda, este trabajo no habría salido adelante. Les agradezco su constante apoyo y su paciencia. En tiempos como los que corren, que aún existan investigadores dispuestos a transmitir su conocimiento y experiencia, no tiene precio, pese a las tijeras que constantemente penden sobre la investigación y el desarrollo.

Gracias también a mis compañeros de trabajo, a los que tengo la fortuna de también poder considerar mis amigos. Sin Pedro, Nelson, Kilian y Airam, estoy completamente seguro de que las cosas no hubiesen sido iguales. No puedo olvidarme de Carlos, compañero en mi anterior trabajo y que compartió conmigo la primera parte del máster, hasta que terminó yéndose al extranjero como tantos compañeros actualmente. Con Jose han sido muchas las tardes, muchos los cafés y las conversaciones sobre el sentido de la vida, el universo y todo lo demás = 42.

También agradezco el interés de todos mis amigos, que en un momento u otro se han preocupado por cómo me iban las cosas: Belén, Natalia, Paula, Lourdes, Elena, Mara, Antonio, Víctor, Tewise... y tantos otros que posiblemente olvide nombrar.

Por último, y no por ello menos importante, me gustaría dar las gracias al programa de Becas y Ayudas al Estudio del Ministerio de Educación, Cultura y Deporte. Sin su aporte económico no habría podido sufragar los gastos de los estudios de máster.

Recurriré finalmente al álgebra, para así dar las gracias al conjunto complementario. Es la única forma de estar completamente seguro de no olvidar a nadie.

**GRACIAS**

IUSIANI. Universidad de Las Palmas de G.C.

“Ahí Señor Sulu, hay una distorsión, ¿la ve?”

**James T. Kirk** en *Star Trek III. En busca de Spock*.

IUSIANI. Universidad de Las Palmas de G.C.

---

# Índice

<b>1. Introducción y estado actual del tema</b>	<b>5</b>
<b>2. Fundamentos Teóricos</b>	<b>8</b>
2.1. Transformada de Hough . . . . .	8
2.1.1. Detección de líneas rectas . . . . .	8
2.1.2. Modificaciones a la transformada de Hough . . . . .	10
2.2. Modelo de distorsión radial . . . . .	13
2.2.1. Estimación del modelo de distorsión . . . . .	15
2.2.2. Corrección de la distorsión . . . . .	18
2.3. Detección de contornos . . . . .	19
2.3.1. Método sub-píxel . . . . .	19
2.3.2. Método de Canny . . . . .	23
<b>3. Desarrollo</b>	<b>25</b>
3.1. Método sub-píxel . . . . .	25
3.2. Método de Canny . . . . .	25
3.3. Corrección de la distorsión . . . . .	28
<b>4. Resultados</b>	<b>31</b>
4.1. Detectores de contornos . . . . .	31
4.1.1. Sub-píxel . . . . .	31
4.1.2. Canny . . . . .	33
4.2. Imágenes sintéticas . . . . .	37
4.3. Imágenes reales . . . . .	39
4.4. Líneas de Hough en función del detector de bordes . . . . .	40

**5. Conclusiones y Líneas Futuras**

**46**

## Índice de figuras

1.	Modelo proyectivo simple . . . . .	5
2.	Representación de la recta . . . . .	9
3.	Representación de la recta en el espacio de Hough . . . . .	10
4.	Nuevo sistema de votación . . . . .	12
5.	Diferencia entre la orientación corregida y la del contorno . . . . .	12
6.	Tipos de distorsión radial . . . . .	14
7.	Modelo de distorsión radial . . . . .	14
8.	Estimación modelo de distorsión . . . . .	18
9.	Efecto parcial 2D . . . . .	20
10.	Ventana en una imagen suavizada . . . . .	21
11.	Resultado sub-píxel sobre imagen médica . . . . .	23
12.	Diagrama de flujo método sub-píxel . . . . .	26
13.	Diagrama de flujo del método de Canny desarrollado . . . . .	27
14.	Sub-píxel vs Canny . . . . .	28
15.	Diagrama de flujo de la corrección iterativa de la distorsión . . . . .	29
16.	Resultado sub-píxel . . . . .	32
15.	Resultado Canny . . . . .	36
16.	Imagen sintética . . . . .	37
17.	Fotografía patrón de calibración . . . . .	38
18.	Estantería . . . . .	39
19.	Estación de metro . . . . .	41
20.	Cuadro . . . . .	42

20.	Evolución líneas de Hough con sub-píxel . . . . .	44
21.	Evolución líneas de Hough con Canny . . . . .	45

.

---

# 1. Introducción y estado actual del tema

La distorsión de lentes es un fenómeno presente en las cámaras, ya sean de vídeo o fotografía, especialmente en las provistas de lentes tipo gran angular o *zoom*. Este efecto viene determinado por la propia geometría de la lente y por la variación que se realiza sobre la denominada distancia focal. Esta variación tiene influencia en cómo la luz pasa a través de la lente y se proyecta en el sensor del dispositivo. Es deseable corregir este tipo de distorsión presente en las imágenes resultantes de la captura, ya que en algunas aplicaciones de Visión por Computador, como por ejemplo la reconstrucción tridimensional a partir de pares estéreo, esta distorsión afecta negativamente a los resultados.

En el modelo proyectivo simple (*pinhole model*) un punto en el espacio 3D  $(x, y, z)$ , la escena, se proyecta en la imagen en un punto 2D  $(x, y)$ , el sensor, en coordenadas proyectivas (figura 1). Las características de una cámara se establecen a través de una serie de parámetros, divididos principalmente en dos tipos: intrínsecos y extrínsecos. Los parámetros intrínsecos vienen determinados por el alto y el ancho de los píxeles de la imagen que se captura (*pixel aspect ratio*), la distancia focal (que puede ser variada mediante el mecanismo de *zoom*) y el punto principal. El foco de la cámara suele estar centrado respecto al negativo o, en el caso de cámaras digitales, respecto al sensor CCD en el que se proyecta la captura. Los parámetros extrínsecos de una cámara vienen caracterizados generalmente por una matriz de rotación y el foco. Estos parámetros determinan la posición y orientación de la cámara en el espacio. A diferencia de los parámetros extrínsecos, los intrínsecos son internos a la cámara y no dependen de su posición en el espacio.

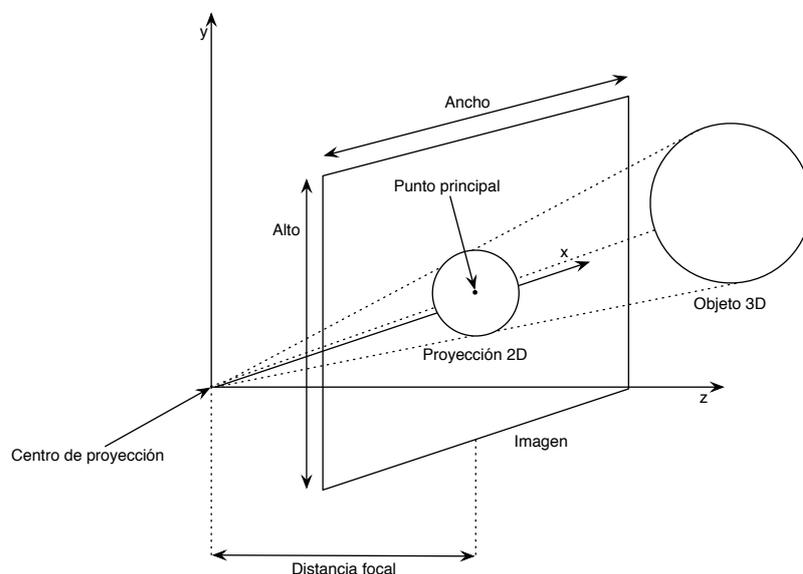


Figura 1: Modelo proyectivo simple (*pinhole model*)

Existen diversos trabajos sobre la estimación y corrección de la distorsión de lentes. La mayoría pasa por un modelado de la misma, que posteriormente permita aplicar algún tipo de proceso corrector sobre la imagen de entrada. Generalmente es preciso indicar primitivas geométricas en la imagen, para así conocer cuál es la forma real que tendría algún objeto

presente en la escena. Una forma puede ser que se marquen aquellas partes que son rectas en la escena real, pero que en la imagen resultante de la captura presentan algún tipo de distorsión.

La rectificación de la imagen de entrada pasa por la estimación de un modelo de distorsión a partir de información de la imagen de entrada, como puede ser la que se ha descrito. En [2] se usa una aproximación general para aproximar el polinomio de un modelo de distorsión radial mediante la imposición de que la proyección de líneas de la escena tridimensional en el plano bidimensional sean líneas rectas. A partir de ello se propone una nueva técnica rápida, mediante la cual se obtiene el modelo del parámetro de distorsión usando una nueva medida del error de la distorsión de la lente, basada en la desigualdad de Cauchy-Schwarz.

La mayoría de técnicas para obtener el modelo de distorsión no toman en cuenta que dicha distorsión depende de la profundidad de un punto de la escena respecto al plano de proyección de la cámara. En [4] se propone un nuevo modelo de distorsión de la lente que depende de la profundidad, proporcionándose una técnica que permite estimar un nuevo modelo computacional de la lente de la cámara, dando un mayor nivel de precisión de calibración.

Otro factor a tener en cuenta son aquellas lentes que permiten hacer *zoom*. En algunos trabajos como en [1], se propone el uso de nuevos modelos para estudiar la variación de la distorsión de la lente cuando se modifica la distancia focal mediante el *zoom*. Estos modelos se basan en una aproximación polinómica que permite tomar en cuenta la variación de la distorsión radial cuando se varía el *zoom*. Para ello se usa la minimización de una función de energía del error global, midiendo la distancia entre secuencias de puntos distorsionados alineados y líneas rectas después de la corrección de la distorsión.

Para la estimación de todos estos modelos es precisa la localización de primitivas en la imagen de entrada para poder realizar una estimación del modelo de distorsión. En relación a ello, en [6] se presenta una transformada de Hough con un esquema de votación mejorado para la detección de líneas en tiempo real. La transformada de Hough es una herramienta potente para la detección de líneas rectas en una imagen, así como otro tipo de curvas como circunferencias o elipses, debido a su resistencia al ruido y a la pérdida de información. Permite la detección de líneas rectas a partir de una serie de puntos de interés de la imagen de entrada.

El método propuesto evita que un punto vote a todas las rectas posibles, identificando grupos de segmentos colineales. Para cada uno de dichos puntos, se limita la región de votación mediante el uso de una pequeña región gaussiana de base elíptica. Esto permite restringir la participación sólo a unas pocas líneas del espacio de parámetros, dependiendo de la calidad del ajuste de sus píxeles a una línea recta. Con ello se consigue una aceleración dentro del proceso de votación de todos los puntos al conjunto de posibles rectas que pueden pasar por ellos.

Por tanto, si se añade un mecanismo que permita aglutinar puntos de interés dentro de la imagen en determinadas primitivas, esto permitiría la estimación del modelo de distorsión. De esta manera, se automatizaría el proceso de selección de regiones rectas en la escena real 3D que al ser proyectadas en plano 2D pueden presentar cierta distorsión de la lente, evitando que haya que indicarse de manera manual.

La estimación inicial de las líneas en la imagen a partir de los puntos de interés es probable

---

que no sea perfecta. Es decir, inicialmente esta estimación será dependiente del grado de distorsión presente. Por ello, las rectas que se encuentran en la imagen de entrada (curvadas por la distorsión de la lente) comienzan por ser estimadas como conjuntos de rectas, ya que no es posible aproximar una curva a una sola recta. Durante el proceso propuesto, se irá realizando la corrección y proporcionando como entrada a la siguiente fase la imagen resultante de la corrección de la etapa previa. Por esta razón puede decirse que se trata de un sistema iterativo incremental de corrección de la distorsión de la lente.

El presente trabajo se estructura en las siguientes secciones:

- **Fundamentos Teóricos:** en esta sección se describen las herramientas usadas, tanto para la detección de los puntos de interés como para la posterior agrupación de estos en primitivas. En este sentido, además de describirse la transformada de Hough, se habla también de algunas modificaciones realizadas a la misma. Dichas modificaciones tienen como objetivo mejorar el proceso de detección a través de cambios en el sistema de votación y en la corrección de la orientación de la tangente del contorno. Como sistemas de detección de puntos de interés se hablará principalmente de dos técnicas: el método sub-píxel [13][14] y el método de Canny [5].
- **Desarrollo:** en este apartado se describen los métodos desarrollados y su funcionamiento. Para ello se apoyará la descripción en los diagramas de flujo que explican el comportamiento de cada una de las partes del sistema implementadas.
- **Resultados:** una vez se conocen las herramientas usadas y su implementación, es el momento de probar su funcionamiento. En primer lugar se hará un estudio acerca de los métodos de detección de contornos que se usan para la obtención de puntos de interés en la imagen de entrada. Para ello se observará la variación del resultado respecto a la modificación del parámetro que permite el cálculo de los umbrales necesarios. Tras ello se verá el funcionamiento del sistema de corrección de la distorsión de la lente, tanto sobre imágenes sintéticas como en imágenes reales, y variando el detector que proporciona los contornos.
- **Conclusiones y líneas futuras:** en esta sección se pasará a indicar cuáles son las conclusiones que se obtienen del trabajo desarrollado, así como las líneas futuras que se podrían desarrollar a partir de él.

## 2. Fundamentos Teóricos

En esta sección se describirán los fundamentos teóricos necesarios a lo largo del desarrollo del presente trabajo. En la primera parte se describe el concepto de la transformada de Hough, su fundamento y aplicaciones. Asimismo se describirán algunas modificaciones que se han hecho sobre la misma, en relación al peso de votación y a la orientación de la tangente a los contornos. Posteriormente se hablará del modelo de distorsión radial, necesario para la estimación y posterior corrección de la distorsión producida por la lente en la imagen.

Por último, debido a que la transformada de Hough precisa de un conjunto de puntos significativos de la imagen, estos se obtendrán a partir de dos métodos de detección de contornos: el método sub-píxel y del método de Canny. Ambos métodos también serán descritos en esta sección.

### 2.1. Transformada de Hough

La transformada de Hough es una herramienta que permite la detección de primitivas en una imagen. Se denominan primitivas a aquellos elementos gráficos presentes en la imagen, ya sean puntos, rectas, circunferencias o polígonos. En el caso de la transformada de Hough básica, las primitivas detectadas son rectas.

La principal ventaja del uso de esta técnica es que es bastante robusta ante la presencia de ruido en la imagen, así como a la existencia de huecos en la frontera de los objetos.

Para poder ser aplicada, la transformada de Hough necesita una imagen binaria de los píxeles pertenecientes a los contornos de la imagen. Tradicionalmente, para este proceso se hace uso del operador gradiente ( $\nabla$ ) y de un proceso de umbralización. En el presente trabajo el método usado para la obtención de los puntos de interés es diferente. En nuestro caso se ha usado un método de detección de contornos con precisión sub-píxel, que no sólo da información respecto a la posición del contorno, si no además del salto de intensidad y de la orientación del mismo. Este método se encuentra descrito en la sección 2.3. Además, en esa misma sección se explica otro método alternativo para la detección de estos puntos de interés, el método de Canny.

Una vez se cuenta con un mapa de los puntos de interés de la imagen, se procede a la detección de las rectas de las cuales forman parte dichos puntos.

#### 2.1.1. Detección de líneas rectas

Consideremos la ecuación explícita de la recta (1).

$$y = ax + b \tag{1}$$

en dicha ecuación  $a$  es la pendiente y  $b$  es la ordenada en el origen. El objetivo es encontrar diferentes grupos de puntos alineados que puedan existir en la imagen de entrada. Es decir, los

puntos de la imagen que satisfacen la ecuación de la recta para los distintos valores de  $a$  y  $b$ .

Para una recta dada, los parámetros  $a$  y  $b$  son constantes, mientras que  $(x, y)$  son variables. Supongamos que no se saben los parámetros de la recta pero si se conocen las coordenadas de uno o varias puntos de ésta. Si éste es el caso, se pueden considerar los parámetros como las variables y las coordenadas como las constantes. Siendo así, la ecuación (1) se puede poner de la forma:

$$b = -ax_i + y_i \quad (2)$$

donde  $(x_i, y_i)$  son las coordenadas de un píxel dado.

Si se representa la ecuación (2) para los valores de  $a$  y  $b$  se obtiene una nueva recta donde  $(a, b)$  son ahora las variables. Esta transformación entre el plano imagen (coordenadas  $(x, y)$ ) y el plano o espacio de parámetros (coordenadas  $(a, b)$ ) es a lo que se denomina *Transformada de Hough* (fig. 2).

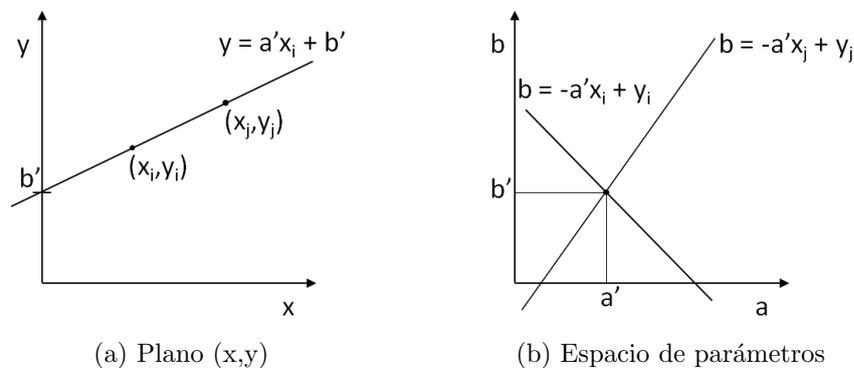


Figura 2: Representación de la recta

En dicho espacio de parámetros (también llamado espacio de Hough), cada punto  $(x, y)$  del plano imagen se convierte en una recta de pendiente  $-x$  y ordenada en el origen  $y$ , que representa el lugar geométrico de todas las rectas que en el plano imagen pasan por el punto  $(x, y)$ . Por lo tanto, la intersección en el espacio de parámetros de dos rectas  $(x_i, y_i)$  y  $(x_j, y_j)$  da lugar al par  $(a', b')$  correspondiente a la recta que pasa por los puntos  $(x_i, y_i)$  y  $(x_j, y_j)$ . De esta manera, puntos alineados en el plano se transforman en el espacio de Hough en sendas rectas que se cortan en un único punto  $(a', b')$ .

Para aplicar la transformada de Hough es preciso discretizar el espacio de parámetros en una matriz de acumuladores. Esta discretización se realiza sobre los intervalos  $(a_{min}, a_{max})$  y  $(b_{min}, b_{max})$  que definen el conjunto de rectas buscadas. A esta estructura se le suele denominar acumulador y viene a estar representada por una matriz en cuyas posiciones se encuentra la cantidad de puntos que recibe cada recta a la que está asociada dicha posición (la posición representa a una recta con un determinado ángulo y orientación).

El siguiente paso es evaluar la ecuación (2) para cada punto  $(x_k, y_k)$  de la imagen y cada uno de los valores de  $a$  en el intervalo  $(a_{min}, a_{max})$ . Los valores de  $b$  obtenidos se redondean al valor discreto más cercano. Así, si al tomar  $a_i$  se obtiene  $b_j$ , se incrementa en uno el valor de votos de la celda  $A(i, j)$ , es decir,  $A(i, j) = A(i, j) + 1$ .

Cuando finaliza este proceso, el número de votos obtenidos en cada celda indica el número total de puntos que satisfacen la ecuación de la recta correspondiente. Por tanto, las celdas con mayor valor del acumulador de votos forman el conjunto de rectas detectadas.

Un problema que aparece al usar la ecuación (1) para representar una recta es que, tanto la pendiente como la ordenada en el origen tienden a infinito conforme la recta se acerca a posiciones verticales. Para evitar este problema se usa la representación normal de una recta dada por:

$$x \cos \theta + y \operatorname{sen} \theta = d \quad (3)$$

donde  $d$  es la distancia de la recta al origen y  $\theta$  es el ángulo que forma la normal con el eje  $x$  (fig. 3).

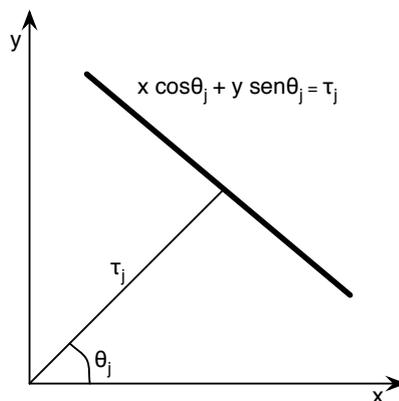


Figura 3: Representación de la recta en el espacio de Hough

La forma de construir el acumulador en el plano  $\theta - d$  es similar al descrito anteriormente, aunque para cada punto que pertenece a la recta se obtiene en el plano  $\theta - d$  una senoide en lugar de una recta como ocurría con la representación explícita.

Como normalmente los puntos provienen de la umbralización de una imagen gradiente, la posición de éstos puede verse afectada de error (debido a ruido, artefactos, efectos espurios, iluminación, etc.). Por ello, las celdas de mayor acumulación no van a aparecer de forma aislada en el espacio de parámetros, sino que se encontrarán formando parte de agrupaciones de celdas con elevados votos.

Todo este procedimiento para la detección de rectas usando la transformada de Hough puede simplificarse si se toma en cuenta la dirección en cada punto. De esta forma se restringe el rango de orientaciones a la hora de evaluar el punto[8].

### 2.1.2. Modificaciones a la transformada de Hough

A pesar de que en el sistema desarrollado se hace uso de la transformada de Hough, ésta ha sido modificada ligeramente. Estos cambios comprenden principalmente dos aspectos dife-

renciados: la forma en que un punto vota a una recta y la inclusión de la corrección de la orientación.

### Sistema de votación

Anteriormente se ha descrito la forma en la que un punto contribuye a una recta en el espacio transformado de Hough. De forma breve, el punto vota a una recta que viene representada por un ángulo y una distancia. Generalmente, el ángulo varía entre  $0^\circ$  y  $180^\circ$ , puesto que para ángulos superiores las rectas comienzan a repetirse. La distancia máxima a la que una recta podrá encontrarse será la longitud de la diagonal de la imagen ( $\sqrt{width^2 + height^2}$ ). Estas dimensiones determinan el ancho y alto del acumulador, es decir, la estructura en la cual se va guardando la aportación de cada uno de los puntos a las diferentes rectas.

Lo normal es que los puntos no coincidan exactamente sobre la recta del espacio de Hough, así que se considera una cierta vecindad. Sin embargo, la aportación que cada punto hace a una recta suele ser de una unidad. Es decir, cada punto emite un voto cuyo valor es uno y que incrementa una posición determinada en el acumulador (representada por el par distancia, ángulo). Independiente de la distancia a la que se encuentre el punto en cuestión con respecto a la recta, su aportación en la acumulación de votos será exactamente igual que un punto que se encontrase justo sobre la misma.

La primera variación propuesta para la transformada de Hough contempla el sistema de votación. No es lógico que, aunque se explore una cierta vecindad, cada punto aporte el mismo peso independientemente del lugar en el que se encuentre con respecto a la recta en cuestión. Es por ello que se propone una variación en el voto emitido, de manera que este sea dependiente de una variable distancia. Esta distancia, al contrario que la descrita anteriormente, no representa a la distancia que se encuentra la recta del origen de coordenadas del espacio transformado de Hough. La distancia considerada al votar un punto es aquella a la que el mismo se encuentra de la recta. En la ecuación (4) se expresa la nueva forma en que un punto contribuye a una recta determinada en el espacio de Hough.

$$vote_{(x,y)} = \frac{1}{1 + d_{\rho,\alpha}} \quad (4)$$

Un punto  $(x, y)$  votará a una recta representada por el par  $(\rho, \alpha)$  en función de la distancia  $d_{\rho,\alpha}$  a la que se encuentre de dicha recta. Si el punto se encuentra encima de la misma, su aportación a la votación será máxima, con un valor de 1. A medida que la distancia del punto a la recta aumenta, la aportación irá decreciendo. Para el cálculo de la distancia se procede de la siguiente manera: Si la ecuación de la recta viene dada por  $\cos(\alpha) \cdot x + \sin(\alpha) \cdot y + d = 0$ , entonces la distancia de un punto  $(x, y)$  a dicha recta viene dada por  $|\cos(\alpha) \cdot x + \sin(\alpha) \cdot y + d|$ .

En la figura 4 se puede ver cuál es el efecto de esta modificación. A la izquierda de la figura se encuentra una captura del acumulador de votaciones. Puede observarse que el punto central tiene un color más oscuro y que conforme se alejan de él los valores son inferiores, ya que se está a más distancia de la recta. A la derecha de la misma figura aparece una representación tridimensional de esta votación. Puede verse como existe cierto comportamiento gaussiano en

la acumulación de las votaciones en la región cercana a cada recta.

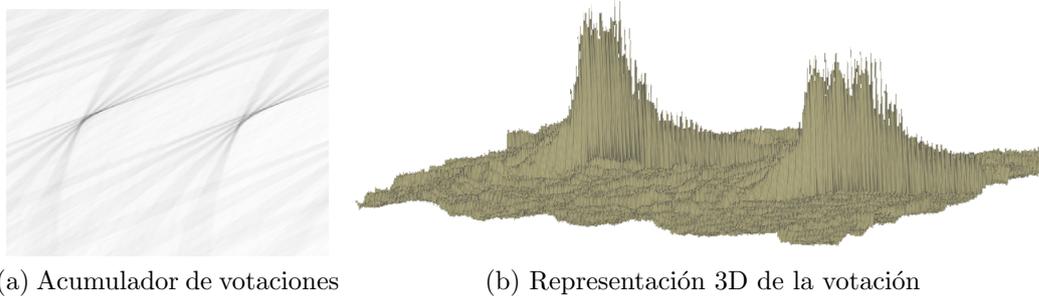


Figura 4: Nuevo sistema de votación basado en la distancia del punto a la recta en el espacio de Hough

### Corrección de la orientación

Previamente se ha descrito que una recta que pase por un punto en el espacio de Hough puede tener una orientación que va desde los  $0^\circ$  hasta los  $180^\circ$ . Cuando un punto vota por una recta ha de considerar la orientación que ésta puede tener. Recorrer todo el conjunto de orientaciones es bastante lento. Este efecto puede mitigarse estableciendo una determinada resolución a la hora de barrer el conjunto de orientaciones, pero se puede correr el riesgo de saltarse las correctas. En la sección 1 se hablaba de que en [6] se proponía una nueva implementación para la detección de rectas en tiempo real mediante el uso de la transformada de Hough. La aceleración del proceso se conseguía mediante la limitación de la región de votación definiendo una región gaussiana de base elíptica, restringiendo así el proceso sólo a un determinado conjunto de rectas.

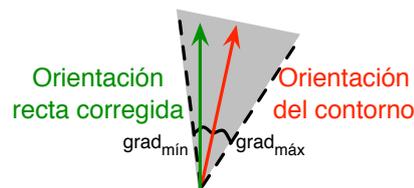


Figura 5: Diferencia entre la orientación corregida y la del contorno

Inicialmente en el sistema propuesto se impone que las rectas a las que los puntos votan no pueden estar más allá de una diferencia de un número determinado de grados respecto a la orientación del contorno. Consideremos una situación similar a la de la figura 5. A partir del método de detección de contornos se tiene la orientación del contorno. Sabiendo esto, la recta que pase por ese punto sólo podrá diferir en la orientación en un corredor que va de  $grad_{min}$  a  $grad_{max}$ . Esto permite, por un lado, acotar la búsqueda de la recta a un rango determinado. Por otro, esta acotación hace que el sistema sea más rápido ya que no se recorren todas las orientaciones posibles.

No obstante a pesar de esta acotación, se está intentado ajustar los puntos a una recta. Esto supone que el número de grados de libertad que haya que recorrer sea mayor que el que

podría suponerse, puesto que es necesario recorrer un rango superior que en el caso en el que la distorsión estuviera corregida.

Por ello se propone también una corrección en la orientación del contorno. Para ello se estima un modelo el cual se aplica sobre la orientación actual. Esto permite que la orientación del contorno original se ajuste mejor el contorno que se está buscando, haciendo que la votación sea más correcta puesto que el contorno que se esta evaluando será más cercano a la recta. Por otro lado, también permite limitar aún mas los grados de libertad, ya que la orientación del contorno y de las rectas buscadas estarían ahora más cerca, haciendo así que el sistema de votación vaya algo más rápido.

Sea  $(x, y)$  perteneciente a un borde y  $(\cos(\alpha), \sin(\alpha))$  la orientación de dicho contorno. Entonces  $(x + \cos, y + \sin)$  será un punto en la dirección de la tangente al contorno a distancia 1. La corrección del contorno consiste en aplicar el modelo a ambos puntos,  $(x, y)$  y  $(x + \cos, y + \sin)$  tal que se obtienen dos nuevos puntos:  $(x', y')$  y  $(x'', y'')$  respectivamente. La diferencia normalizada entre estos dos nuevos puntos supone la corrección en la orientación que es necesaria realizar.

## 2.2. Modelo de distorsión radial

Las lentes, al igual que otros sistemas ópticos, tienen defectos que pueden producir que la imagen se vea borrosa, cambios en el color, distorsiones geométricas, etc. Estos defectos se denominan normalmente en óptica como aberraciones. Los rayos deben pasar a través de algún medio, como el aire, el cual también puede causar que se vea un efecto borroso.

En un sistema óptico ideal, cada punto en el objeto será proyectado en un único punto de tamaño cero en la imagen. En realidad, un punto en el objeto real no es un punto en el plano de la imagen. El resultado es una distribución de intensidades en el volumen el cual no tiene forma simétrica. En la práctica, la distorsión es tanto mayor cuanto menor es la distancia focal<sup>1</sup>, es decir, cuanto mayor es el ángulo de visión.

En aplicaciones de visión por computador la distorsión radial es una aberración que tiene que ser estimada y corregida. Una lente típica tiene una distorsión de algunos píxeles que no son visibles para un observador cuando se observa una escena natural. Sin embargo, cuando una imagen se usa para tomar medidas, es necesaria la compensación de esa distorsión.

El modelo práctico de una aberración geométrica en una lente incluye dos componentes de distorsión. El primero es la **distorsión radial** la cual es causada porque la lente proyecta el rayo más o menos que en un caso ideal. El segundo es el **desplazamiento del punto principal** con respecto al punto central de la imagen.

La distorsión de cojín o de barril están centradas y son simétricas (fig. 6). Para aproximar la corrección necesaria se usa un modelo matemático simple. Se trata de un polinomio de orden

---

<sup>1</sup>La distancia focal es la distancia entre el centro óptico de la lente y el foco del sensor donde se proyecta. El efecto comentado se da sobre todo en lentes de tipo gran angular (*wide-angle lens*).

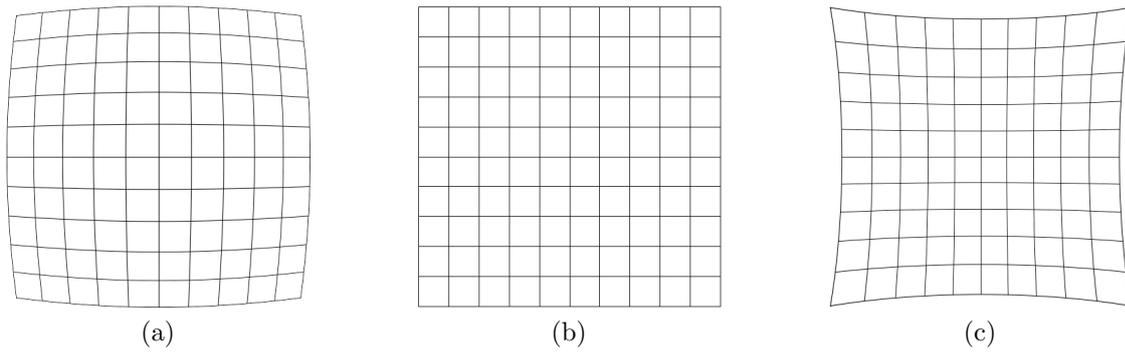


Figura 6: Distorsión radial realizada sobre un grid de  $10 \times 10$ : (a)distorsión tipo barril, (b)imagen original y (c)distorsión tipo cojín

inferior dependiente de la distancia  $r$  del punto observado  $(x', y')$  respecto del punto principal  $(x_0, y_0)$  de la imagen. Las variables involucradas aparecen en la figura 7. Las coordenadas de un píxel después de la corrección son  $x = x' + \Delta_x$  y  $y = y' + \Delta_y$ .

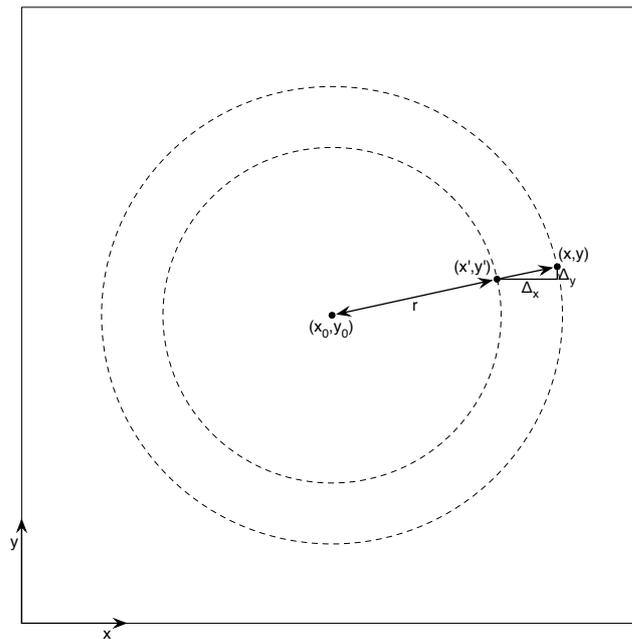


Figura 7: Modelo de distorsión radial:  $(x', y')$  son las coordenadas sin corregir.  $(x, y)$  son las coordenadas del punto después de realizar la corrección.  $(x_0, y_0)$  son las coordenadas del punto principal o centro de distorsión. Finalmente,  $(\Delta x, \Delta y)$  son las componentes de la corrección necesaria y  $r$  es la distancia entre el centro de distorsión y el punto sin corregir.

La distorsión radial (ec. 5) es aproximada como una función simétrica rotacional con respecto al punto principal y dependiente de la distancia  $r$  del píxel medido  $(x', y')$  al punto principal  $(x_0, y_0)$ .

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2} \quad (5)$$

Polinomios de orden inferior que dependen sólo de  $r$  se usan para asegurar la simetría rotacional. Se usan comúnmente polinomios con al menos orden seis. La aproximación es

$$\begin{aligned}\Delta_x &= (x' - x_0)(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \\ \Delta_y &= (y' - y_0)(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots)\end{aligned}\quad (6)$$

La distorsión está representada por los coeficientes  $k_1, k_2, k_3$  que son obtenidos experimentalmente para una lente particular mediante la observación de una imagen de calibración conocida, por ejemplo, cubierta con algunos patrones regulares de estructuras o líneas. A menudo es suficiente utilizar solo coeficientes con valor distinto de cero para el coeficiente  $k_1$  en la aproximación [11].

### 2.2.1. Estimación del modelo de distorsión

En [2] se describe que el modelo básico estándar para la compensación de la distorsión de barril o cojín es un modelo de distorsión radial dado por la siguiente expresión:

$$\begin{pmatrix} \hat{x} - x_c \\ \hat{y} - y_c \end{pmatrix} = L(r) \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix}\quad (7)$$

donde  $(x, y)$  son las coordenadas de los puntos originales (imagen distorsionada),  $(\hat{x}, \hat{y})$  son las coordenadas de los puntos corregidos (imagen corregida),  $(x_c, y_c)$  es el centro del modelo de distorsión de la cámara, normalmente el centro de la imagen,  $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$  y  $L(r)$  es la función que define la forma del modelo de distorsión. Generalmente,  $L(r)$  es aproximada mediante una expansión de Taylor, que es:

$$L(r) = k_0 + k_1 r + k_2 r^2 + k_3 r^3 + \dots$$

donde el conjunto  $\mathbf{k} = (k_0, k_1, \dots, k_{N_k})^T$  son los parámetros de la distorsión. La complejidad del modelo viene dada por número de términos de la expansión de Taylor que se usan para aproximar  $L(r)$ .

Para determinar  $L(r)$  se usa la aproximación general por la que se impone el requisito de que la proyección de líneas 3D en la imagen han de ser líneas rectas 2D.

La distorsión de la lente está incluida en el modelo de calibración de la cámara de la siguiente manera: dada una cámara definida por una matriz de rotación  $R$ , un foco  $c = (c_x, c_y, c_z)^T$  y una matriz  $A$  de  $3 \times 3$  de parámetros intrínsecos, la proyección  $(x, y)$  de un punto 3D  $X = (X, Y, Z)$  en la cámara viene dada por la siguiente expresión:

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix} = sA(R, -Rc) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

donde  $(\hat{x}, \hat{y})$  está definida en la ecuación (7) y  $s$  es valor usual del factor proyectivo. En el caso de  $L(r) \equiv 1$ , el modelo de la cámara está libre de distorsión de lente y la expresión anterior se convierte en el modelo proyectivo “*pinhole*” usual donde la proyección de un punto  $X$  3D en la cámara viene dado por la intersección de la línea  $\overline{cX}$  con el plano retinal. Se observa que la corrección de la distorsión de la lente viene realizada en coordenadas de píxel de la imagen. A fin de que el modelo de distorsión de la lente sea una función radial en coordenadas de píxel, se necesita asumir que el sensor CCD de la cámara tiene píxeles cuadrados. Esta suposición de un tamaño de píxel cuadrado es satisfactoria, en la práctica, para la mayoría de las cámaras digitales modernas. Por otro lado, esta misma suposición puede olvidarse si se usan coordenadas normalizadas en lugar de coordenadas de píxel, pero para normalizar las coordenadas de un punto se necesitan conocer los parámetros intrínsecos de la cámara. Ya que se supone que los parámetros intrínsecos de la cámara son desconocidos, se ha formulado la corrección de la distorsión de la lente en términos de coordenadas de píxel.

Para calibrar la cámara de forma precisa, normalmente se aplica una técnica lineal para obtener una estimación inicial de los parámetros de la cámara y entonces se realiza un ajuste de haz para mejorar la precisión de la estimación de los parámetros. El ajuste de haz está basado en una minimización no lineal donde la medida del error cuadrático entre la posición estimada y la observada de los puntos de la imagen es minimizada. Normalmente el modelo de distorsión se incluye en el parámetro del ajuste de haz. En este sentido, el método propuesto puede ser usado para una estimación rápida inicial del modelo de distorsión en dicho procedimiento.

Los parámetros de distorsión  $\mathbf{k}$  se calculan estableciendo  $k_0 = 1$ . Con el objetivo de obtener puntos corregidos lo más cercanos posible a los puntos distorsionados se estima un factor de *zoom* para minimizar la suma del cuadrado de la distancia entre los puntos distorsionados y los corregidos. Sea  $s$  el factor de *zoom*, se tiene:

$$\begin{pmatrix} \hat{x} - x_c \\ \hat{y} - y_c \end{pmatrix} = sL(r) \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix}$$

se minimiza:

$$H(s) = \sum_{l=1}^N \sum_{i=1}^{N_l} (\hat{x}_{l,i} - x_{l,i})^2 + (\hat{y}_{l,i} - y_{l,i})^2$$

un cálculo simple conduce a

$$H(s) = \sum_{l=1}^N \sum_{i=1}^{N_l} \left( s \sum_{j=0}^{N_k} k_j^n (r_{l,i})^{j+1} - r_{l,i} \right)^2$$

el mínimo de esta función se alcanza en

$$s_{\min} = \frac{\sum_{l=1}^N \sum_{i=1}^{N_l} \sum_{j=0}^{N_k} k_j^n (r_{l,i})^{j+2}}{\sum_{l=1}^N \sum_{i=1}^{N_l} \left( \sum_{j=0}^{N_k} k_j^n (r_{l,i})^{j+1} \right)^2}$$

y finalmente se actualiza el polinomio  $L(r)$  (es decir  $\mathbf{k}$ ) mediante la multiplicación de todos los coeficientes por  $s_{\min}$  (esto es  $k_j^n = s k_j^n \forall j$ ).

Una ventaja interesante de esta aproximación es que la resolución de la imagen corregida es similar a la resolución de la imagen distorsionada original. Esta es una propiedad muy útil si se necesita generar la imagen corregida a partir de la original distorsionada.

El algoritmo para la estimación del modelo de distorsión de la lente se podría dividir en los siguientes pasos:

1. Se calculan los bordes de la imagen usando un algoritmo detector de bordes con precisión sub-píxel.
2. Se seleccionan algunas colección de puntos borde correspondientes a diferentes segmentos 3D rectos, que serán usados para ajustar los parámetros de distorsión.
3. Se inicializa  $\mathbf{k} = (1, 0, \dots, 0)^T$
4. Se elige cualquier par  $p, q \in \mathbb{Z}(1 \leq p, q \leq N_k)$  y se optimizan  $k_p, k_q$ .
5. Se actualiza  $\mathbf{k}$  usando un factor de *zoom* tal que los puntos distorsionados y corregidos estén lo mas cerca posible.

Para calcular los bordes de la imagen se puede usar cualquier algoritmo detector de bordes.

Es bien sabido que cuando se hace uso de métodos algebraicos es mejor usualmente normalizar las coordenadas de los puntos antes de calcular la solución algebraica del problema. Siguiendo esta estrategia, como primer paso, se normaliza los puntos del contorno  $(x_i, y_i)$  usando la transformación

$$x'_{l,i} = \frac{(x_{l,i} - x_c)}{A} \quad y'_{l,i} = \frac{(y_{l,i} - y_c)}{A}$$

donde  $A$  viene dada por

$$A = \sqrt{\frac{\sum_{l=1}^N \sum_{i=1}^{N_l} (x_{l,i} - x_c)^2 + (y_{l,i} - y_c)^2}{2(N_1 + N_2 + \dots + N_N)}}$$

y se calculan los parámetros de distorsión  $k'_i$  para los puntos del borde normalizados  $\{(x'_{l,i}, y'_{l,i})\}_{i=1}^N$ . Finalmente, con el objetivo de recuperar los parámetros de distorsión  $k_i$  para los puntos del contorno originales, se debe tener en cuenta que siguiendo las expresiones anteriores y (7) se tiene que

$$k_j = \frac{k'_j}{(A)^j}$$

En [3] se propone el uso de esta metodología para la obtención de los parámetros de distorsión mediante la minimización de un polinomio de cuarto grado en varias variables.

Como ha sido descrito con anterioridad y como expone el punto dos del algoritmo, se precisa de la selección de líneas rectas 3D de la escena real que permitan realizar el ajuste de los parámetros de distorsión. En la figura 8 se puede ver un ejemplo de la selección de estos puntos. La elección de los segmentos se realiza de forma manual y establece que aquellas regiones seleccionadas son en realidad rectas, aunque en la imagen producto de la captura aparezcan afectadas por la distorsión de la lente que se trata de estimar.

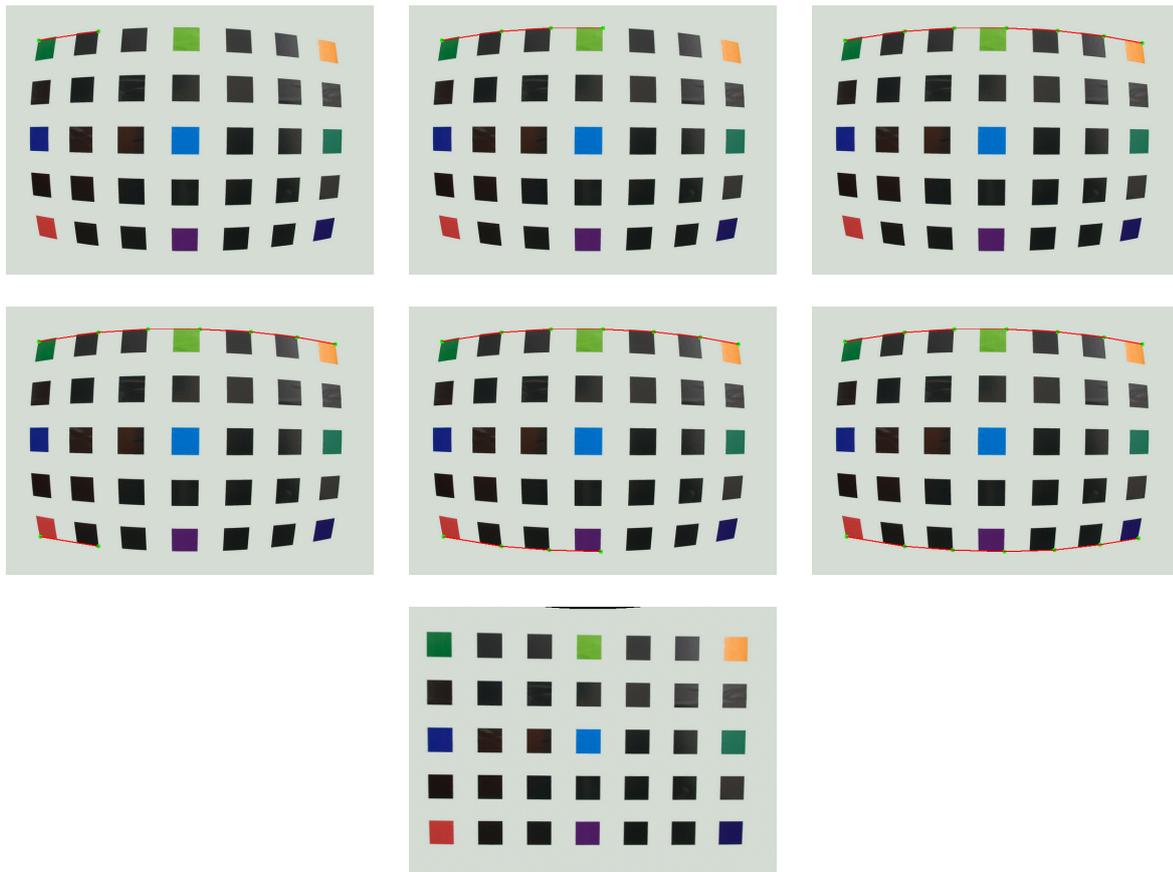


Figura 8: Selección manual de segmentos (que en la escena real son rectos) para la estimación del modelo de distorsión

### 2.2.2. Corrección de la distorsión

Para obtener la imagen libre de distorsión, se precisa calcular la posición correspondiente del píxel en la imagen distorsionada para cada píxel de la corregida. Para determinar la intensidad de los píxeles en la imagen corregida se hace uso de interpolación bicuadrática. Para ello es necesario realizar la inversión del modelo de distorsión radial.

Si se tiene la función radial  $G(\hat{r})$  de la ecuación (8)

$$\begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix} = G(\hat{r}) \begin{pmatrix} \hat{x} - x_c \\ \hat{y} - y_c \end{pmatrix} \quad (8)$$

donde

$$\hat{r} = \sqrt{(\hat{x} - x_c)^2 + (\hat{y} - y_c)^2}$$

De la expresión anterior se obtiene que:

$$r = G(\hat{r})\hat{r}$$

Por otro lado, se tiene

$$\begin{pmatrix} \hat{x} - x_c \\ \hat{y} - y_c \end{pmatrix} = L(r) \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix}$$

y por lo tanto

$$\hat{r} = L(G(\hat{r})\hat{r})G(\hat{r})\hat{r}$$

De esta manera se puede concluir que  $G(\hat{r})$  es una raíz del polinomio

$$P(\lambda) = 1 - L(\lambda\hat{r})\lambda = 1 - \sum_{j=0}^{N_k} k_j \hat{r}^j \lambda^{j+1}$$

Con el objetivo de minimizar la distancia existente entre los puntos distorsionados y los corregidos, se elige, entre todas las raíces reales de  $P(\lambda)$ , aquella más cercana a 1.

## 2.3. Detección de contornos

En la sección 2.1 se decía que era necesaria una imagen binaria de los puntos de interés de la imagen para poder aplicar la transformada de Hough. En ese mismo sentido, se habla de que tradicionalmente se calcula la imagen gradiente de la imagen de entrada y a partir de ella se obtiene la imagen binaria con los puntos de interés a través de proceso de umbralización.

En esta sección se va a describir los dos métodos que han sido usados para la detección de contornos. El primero en ser descrito es una técnica de detección de contornos con precisión sub-píxel. El segundo es un método tradicional en la literatura de detección de bordes, el método de Canny.

### 2.3.1. Método sub-píxel

Un borde se puede definir como una línea virtual entre dos regiones con diferentes niveles de intensidad. Normalmente, los métodos estándar sólo indican cuál es el píxel borde. Para dicho fin se basan en información numérica de una cierta vecindad, y así se decide si se considera borde o no. Además, se asume para ello que la imagen es una función continua y derivable en todo punto. Es decir, aunque la imagen que se use sea una matriz de valores, se estima que es el resultado del muestreo de una determinada señal continua.

El método propuesto por el profesor Agustín Trujillo Pino en [13] plantea algo diferente. En lugar de considerarse que se está ante una función continua y derivable, se plantea que un borde

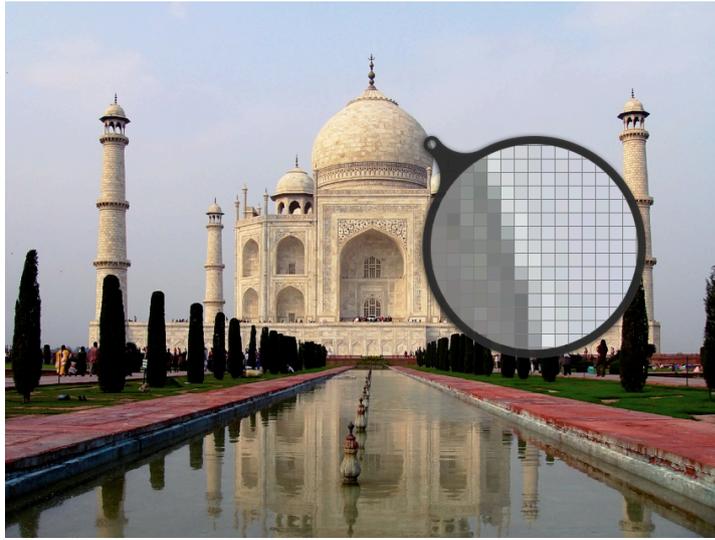


Figura 9: Detalle del efecto parcial producido en un borde

es una discontinuidad en los valores de intensidad de la función, delimitando la frontera entre dos objetos. Cuando un píxel forma parte de un borde tiene una intensidad intermedia, que se encuentra entre la intensidad del objeto y la intensidad del fondo (ver fig. 9). Por tanto, se afirma que la intensidad de un píxel borde es la suma de la contribución de cada color, pesada por el área relativa que ocupa cada uno en el interior de dicho píxel. Es este efecto parcial en el interior del píxel (vóxel en el caso de estar en tres dimensiones) el que permitirá al método localizar de forma precisa el contorno, así como todas sus características: posición sub-píxel, orientación, curvatura e intensidad a ambos lados.

De forma general, las imágenes obtenidas por dispositivos de captura nunca son totalmente ideales, puesto que de un modo u otro se ven afectadas por algún tipo de ruido. Este fenómeno puede afectar a una correcta detección de los contornos. Es necesario procesar la imagen previamente para tratar de eliminarlo en la medida de lo posible.

Si se trata de aplicar un método de detección a una imagen con ruido, el resultado será bastante más caótico, debido a los cambios de intensidad introducidos por el ruido en la imagen. Por esto, al igual que hacen otros métodos como el de Canny, este realiza un suavizado previo de la imagen de entrada.

### Método de Suavizado

El método de suavizado aplicado en la fase previa a la detección es un suavizado de tipo gaussiano, un filtro espacial paso-bajo (a menudo también son denominados como *promediado en el entorno* [7]). Es un método de los denominados isotrópicos (difusión isotrópica), esto es, que el efecto de suavizado se produce de manera uniforme en todas direcciones. La desventaja que causan este tipo de filtros es que no se preservan los contornos (a diferencia de los métodos anisotrópicos (difusión anisotrópica) como los trabajos realizados por Perona y Malik en [10] o por Krissian y Aja-Fernández en [9], por poner dos ejemplos, que se encuentran basados en esquemas de ecuaciones diferenciales en derivadas parciales). No obstante, la solución pasa

porque se plantean las ecuaciones necesarias para que sea posible estimar la posición exacta del contorno, así como el resto de sus características, a pesar de este tipo de filtrado.

Para el suavizado se va usar una máscara de radio 1. Dicha máscara posee la siguiente estructura:

$$H_1 = \begin{pmatrix} a_{11} & a_{01} & a_{11} \\ a_{01} & a_{00} & a_{01} \\ a_{11} & a_{01} & a_{11} \end{pmatrix} \quad (9)$$

donde se cumple que

$$\begin{aligned} 1 &= a_{00} + 4a_{01} + 4a_{11} \\ a_{00} &\geq a_{01} \geq a_{11} > 0 \end{aligned} \quad (10)$$

Al realizar la convolución de la imagen de entrada  $F$  con la máscara  $H_1$  se obtiene la imagen  $G$  tal que

$$G = F * H_1 \quad (11)$$

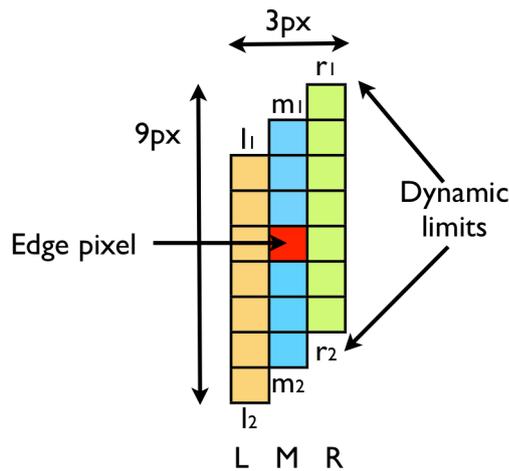


Figura 10: Ventana para el cálculo de los parámetros del contorno en una imagen suavizada [12]

Se demuestra que usando solamente la información de la imagen suavizada  $G$ , independientemente de los valores de  $a_{ij}$  usados en la máscara de suavizado (aunque sabiendo que se trata de una máscara con radio 1), se pueden obtener de forma exacta los parámetros del contorno a través de las ecuaciones que aparecen a continuación. No obstante, se consideran los límites de la ventana para el cálculo de las sumas, puesto que si no los valores que se cogen para realizar la estimación de  $A$  y  $B$  podrían estar muy alejados y tener valores que no serían los deseados para el cálculo. Estos límites serán diferentes en función de la pendiente, por ello se podría decir que se tendría una especie de ventana escalonada (fig. 10).  $S_L$  está desplazada un píxel hacia abajo con respecto a  $S_M$ , y  $S_R$  está desplazada un píxel hacia arriba. Esto se hace para pendientes entre 0 y 1. Sin embargo, para pendientes entre 0 y -1, la situación es la simétrica,  $S_L$  un píxel más arriba que  $S_M$ , y  $S_R$  un píxel más abajo.

Teniendo en cuenta todo lo anterior, a partir de los datos de la imagen suavizada  $G$  se pueden estimar los parámetros del contorno de la imagen original  $F$  (posición, pendiente, magnitud del salto de intensidad y curvatura, medidos sobre la vertical central del píxel), llevando a cabo los siguientes pasos:

En primer lugar se obtienen las sumas de las columnas izquierda, central y derecha de la siguiente manera:

$$\begin{aligned} S_L &= \sum_{k=-3+m}^{3+m} G_{i-1,j+k} \\ S_M &= \sum_{k=-3}^3 G_{i,j+k} \\ S_R &= \sum_{k=-3-m}^{3-m} G_{i+1,j+k} \end{aligned} \quad (12)$$

donde

$$m = \begin{cases} 1 & \text{si } G_x(i, j)G_y(i, j) > 0 \\ -1 & \text{si } G_x(i, j)G_y(i, j) < 0 \end{cases} \quad (13)$$

Posteriormente, se obtiene una estimación para los valores de intensidad a ambos lados del borde:

$$\begin{aligned} B &= \frac{G_{i-m,j-3} + G_{i-m,j-4} + G_{i,j-4}}{3} \\ A &= \frac{G_{i,j+4} + G_{i+m,j+4} + G_{i+m,j+3}}{3} \end{aligned} \quad (14)$$

De esta manera se pueden obtener los coeficientes de la ecuación  $y = a + bx + cx^2$  como:

$$\begin{aligned} c &= \begin{cases} 0 & \text{si se buscan contornos rectos} \\ \frac{S_L + S_R - 2S_M}{2(A-B)} & \text{si se buscan contornos de segundo grado} \end{cases} \\ b &= m + \frac{S_R - S_L}{2(A-B)} \\ a &= \frac{2S_M - 7(A+B)}{2(A-B)} - \frac{1 + 24a_{01} + 48a_{11}}{12}c \end{aligned} \quad (15)$$

Finalmente, se calculan los parámetros del contorno. Se considera que el corte con la vertical está en  $(0, a)$  y el cálculo del vector normal se realiza según la ecuación (16).

$$N = \frac{A-B}{\sqrt{1+b^2}} [b, -1] \quad (16)$$

La curvatura se obtiene como [13]:

$$K = \frac{2cn}{(1+b^2)^{\frac{3}{2}}} \quad (17)$$

donde

$$n = \begin{cases} 1 & \text{si } G_y(i, j) > 0 \\ -1 & \text{si } G_y(i, j) < 0 \end{cases} \quad (18)$$

Existe una extensión de este método en el que se realiza una restauración de la imagen de manera iterativa, mediante la aplicación sistemática del proceso descrito en esta sección, consiguiéndose preservar los contornos detectados. En la figura 11 se observa la precisión del resultado del método frente a una imagen etiquetada manualmente por tres expertos. [14]



Figura 11: Resultado de la aplicación del método vs puntos de referencia segmentados manualmente por expertos

### 2.3.2. Método de Canny

El operador de Canny (1986) está basado en un desarrollo analítico de optimización, partiendo de un modelo continuo unidimensional de un escalón.

Considérese una función escalón con amplitud “ $h_E$ ” afectada por un ruido blanco gaussiano de desviación típica  $\sigma_n$ . Supóngase que la detección de bordes se va a llevar a cabo mediante la convolución de una función continua unidimensional  $f(x)$  con una función respuesta impulsional antisimétrica  $h(x)$ , que tiene amplitud cero fuera del intervalo  $[-W, W]$ . El borde buscado de la función  $f(x)$  se marcará donde aparezca el máximo local del gradiente, obtenido tras la convolución de  $f(x)$  con  $h(x)$ . Para determinar la función  $h(x)$  buscada se exige que ésta satisfaga los siguientes criterios:

- **Buena Detección.** Se maximiza la amplitud de la relación señal-ruido (snr) del gradiente para obtener una alta probabilidad de marcarlo donde no lo hay. La relación señal-ruido para el modelo considerado es:

$$snr = \frac{h_E}{\sigma_n} S(h) \quad (19)$$

con

$$S(h) = \frac{\int_{-W}^0 h(x) dx}{\int_{-W}^W [h(x)]^2 dx} \quad (20)$$

- **Buena Localización.** Los puntos del borde marcados por el operador deben estar tan cerca del centro del borde como sea posible. El factor de localización se define como:

$$LOC = \frac{h_E}{\sigma_n} L(h) \quad (21)$$

con

$$L(h) = \frac{h'(0)}{\int_{-W}^W [h(x)]^2 dx} \quad (22)$$

donde  $h'(0)$  es la derivada de  $h(x)$ .

De acuerdo con estos dos criterios, el detector óptimo de bordes de tipo escalón es un escalón truncado (diferencia de escalones). Este operador, sin embargo, tiende a generar muchos máximos locales en su respuesta a bordes ruidosos de tipo escalón. Aunque estos bordes se deberían considerar erróneos de acuerdo con el primer criterio, sin embargo, no se ha considerado la interacción entre las respuestas en varios puntos próximos. Si se examina la salida del operador de diferencia de escalones, se ve que la respuesta a un escalón con ruido es un pico triangular con varios máximos en la vecindad del borde. Por ello es necesario incorporar el siguiente criterio que corrija este problema.

- **Respuesta Única.** Debe haber una única respuesta para cada borde. La distancia entre picos del gradiente cuando sólo el ruido está presente, denotada por  $x_m$ , se establece como una fracción de  $k$  del ancho del operador, es decir:

$$x_m = kW \quad (23)$$

Canny combina los tres criterios minimizando el producto  $S(h)L(h)$  sujeto a la restricción dada por 23. Debido a la complejidad de esta formulación no existe solución analítica de este problema.

Todo el desarrollo expuesto hasta ahora se refiere a una señal unidimensional continua. En el caso de imágenes digitales (bidimensionales y discreto), el operador propuesto por Canny se aproxima mediante la derivada de la Gaussiana en la dirección perpendicular al borde. En la práctica, aunque existen diferentes implementaciones, los pasos a seguir serían:

1. Calcular el módulo y la dirección del gradiente de la imagen suavizada aplicando el operador DroG (Derivada de la Gaussiana).
2. En la dirección del gradiente, eliminar puntos que no sean máximos locales del módulo (equivalente a encontrar el paso por cero en el operador LoG). Desechando los píxeles que no son máximos locales se mejora la localización y se evitan detecciones falsas.

El proceso de eliminación de no-máximos locales se suele implementar siguiendo el borde en la dirección perpendicular al gradiente, considerando los 8-vecinos.

Normalmente se utiliza una función de histéresis tal que el primer píxel de un segmento del borde debe tener un módulo del gradiente que supere un valor *gradiente\_alto*. Entonces se comienza a añadir píxeles vecinos en la dirección del borde (perpendicular al gradiente) mientras que el valor del módulo de éste no caiga por debajo de un valor *gradiente\_bajo* (menor que *gradiente\_alto*) [8].

---

## 3. Desarrollo

Una vez conocemos los fundamentos teóricos que comprenden este trabajo, llega el instante de hablar sobre el desarrollo de estos métodos y su funcionamiento. Para ello en primer lugar se empezará hablando de los dos métodos de detección de contornos, tanto el sub-píxel como el Canny. Ambos son los que proporcionan posteriormente los puntos de interés necesarios para aplicar la transformada de Hough y así detectar las líneas que permitan estimar el modelo de distorsión de la lente.

### 3.1. Método sub-píxel

Para una mayor claridad, en la figura 12 puede verse el diagrama de flujo del método sub-píxel. Inicialmente se aplica un suavizado sobre la imagen de entrada a través del método descrito en la sección 2.3.

Una vez se ha suavizado la imagen de entrada se entra en un bucle que se encarga de recorrer todos los píxeles de la imagen. Al principio de dicho bucle se calculan las parciales en  $x$  y en  $y$ , para posteriormente ver cuál de las dos es mayor en valor absoluto. En función del resultado de dicha operación lógica se toma una ventana en vertical o en horizontal. Estas ventanas tienen una estructura escalonada para una mejor estimación de los valores de intensidad a ambos lados del contorno. De otra manera los valores tomados para dicho cálculo podrían estar demasiado alejados del píxel borde que se está computando.

Si el valor de la parcial supera el valor del umbral establecido, se pasa a realizar el resto de los cálculos. Para estimar el valor adecuado para el cálculo, se usa un porcentaje. A partir de la imagen del módulo del gradiente se obtiene un valor de umbral que es aquel valor correspondiente de hacer la mediana sobre los valores de dicha imagen dejando a la izquierda  $k$  valores. Por ejemplo, si  $k = 0,9$  se estarían dejando a la izquierda en el cálculo de la mediana el 90 % de los píxeles, considerándose para la aplicación del método solamente el 10 % restante.

Posteriormente se calculan las sumas de las filas/columnas de la ventana y a partir de ello se obtienen los parámetros del contorno. Una vez finalizado el proceso se tendrían, para todos los píxeles borde, su posición sub-píxel, su orientación y el valor de la intensidad a ambos lados del contorno.

### 3.2. Método de Canny

Al igual que se hiciera para describir el flujo del método anterior, se adjunta la imagen del diagrama de flujo del método de Canny (fig. 13).

Inicialmente se calcula a partir de la imagen de entrada una imagen suavizada, fruto de realizar una convolución con una gaussiana. Después se calcula el módulo del gradiente del suavizado. Estos dos pasos conforman el operador DroG (“Derivative of the Gaussian”). El paso

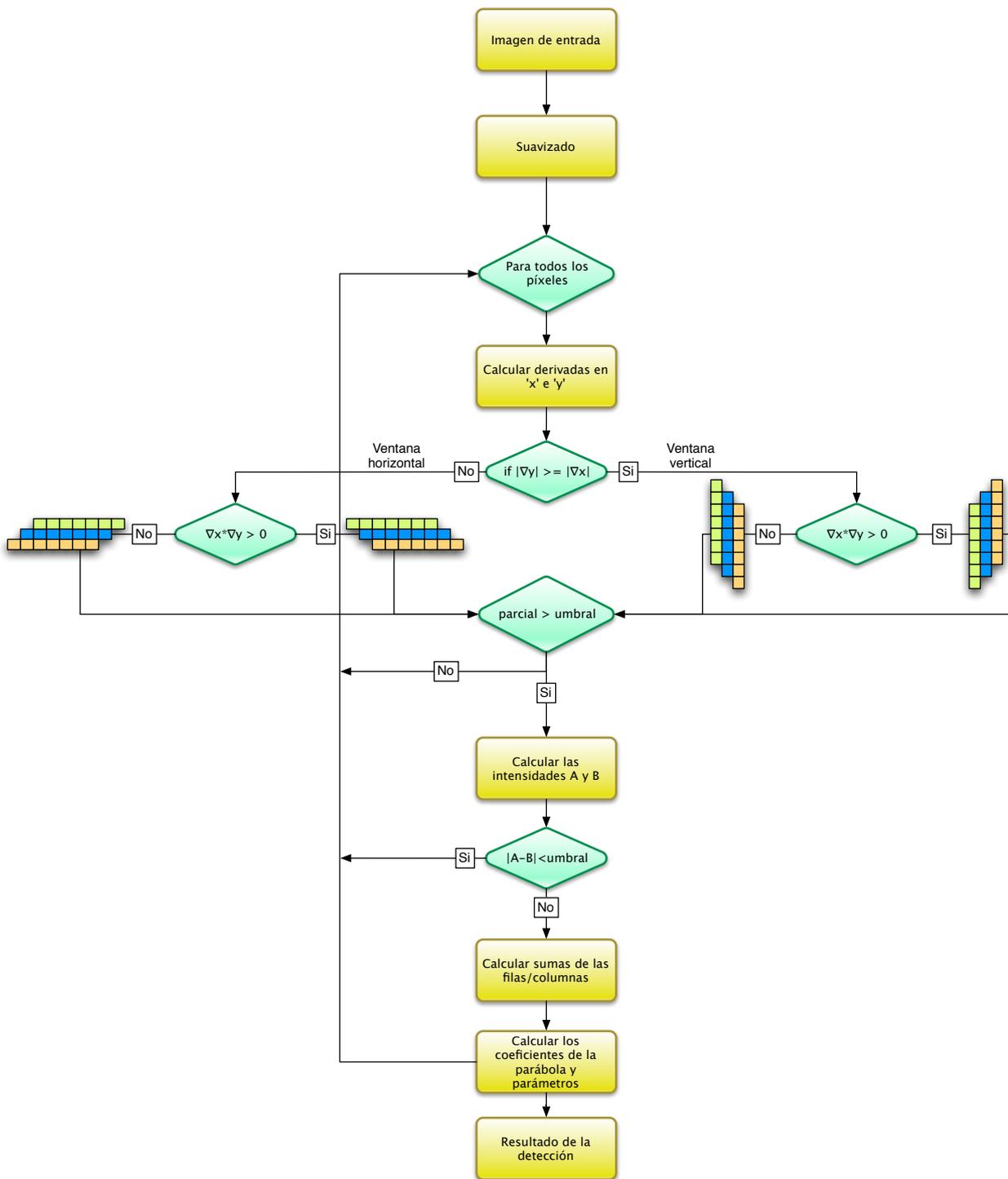


Figura 12: Diagrama de flujo del método sub-píxel de orden 2 para imágenes con ruido

siguiente es la realización del supresión de no máximos. Para ello, el método de Canny previamente discretiza cada píxel en cuatro orientaciones principales. En función a ellas visitará a sus dos vecinos en dicha dirección, determinando si el píxel en cuestión es máximo en su valor del módulo del gradiente. De esta manera se eliminan aquellos píxeles que no dan la información más significativa del borde, eliminando aquellos que no.

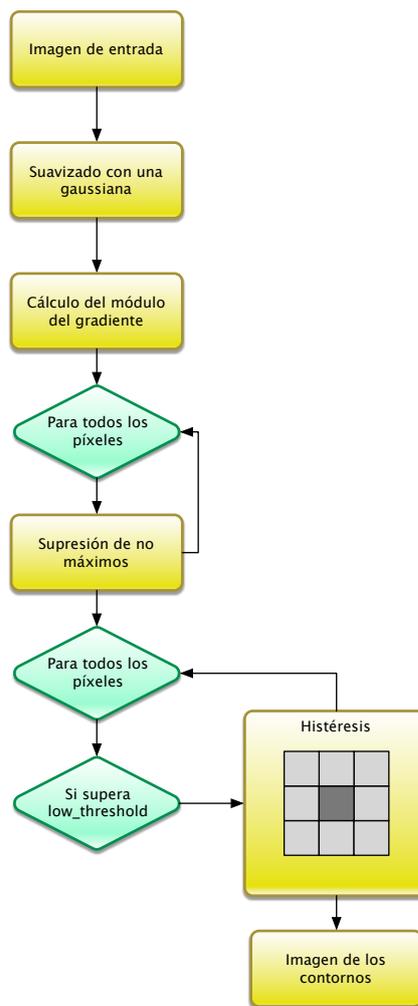


Figura 13: Diagrama de flujo del método de Canny desarrollado

Una vez se ha realizado la supresión de no máximos, se pasa a ejecutar el proceso de histéresis. Si un píxel supera el valor inferior de umbral, se va recorriendo sus vecinos de forma recursiva, marcando todos aquellos que cumplen que su valor es inferior al valor máximo de umbral. De esa forma se tiene finalmente todos aquellos píxeles borde que son máximos en su orientación y que además cumplen que se encuentran entre los valores mínimo y máximo de los umbrales. De la misma forma que se hiciera en el caso anterior, los umbrales se seleccionan a partir de unos porcentajes que se usarán para el cálculo de la mediana dejando a la izquierda  $k$  elementos. La única condición que hay que cumplir es que el umbral inferior sea menor que el superior. Además de la imagen con los contornos también se devuelven las orientaciones, a partir de las componentes  $x$  e  $y$  del gradiente, así como las posiciones de los píxeles borde.

En la figura 14 se puede ver una comparativa entre el resultado proporcionado por el método sub-píxel comparado con el de Canny. A la hora de obtener el umbral en el primero de los métodos se ha usado un valor de  $k = 0,95$ . Es decir, para la selección del umbral se calcula la mediana de la imagen del módulo del gradiente, dejando a la izquierda el 95 % de los valores.

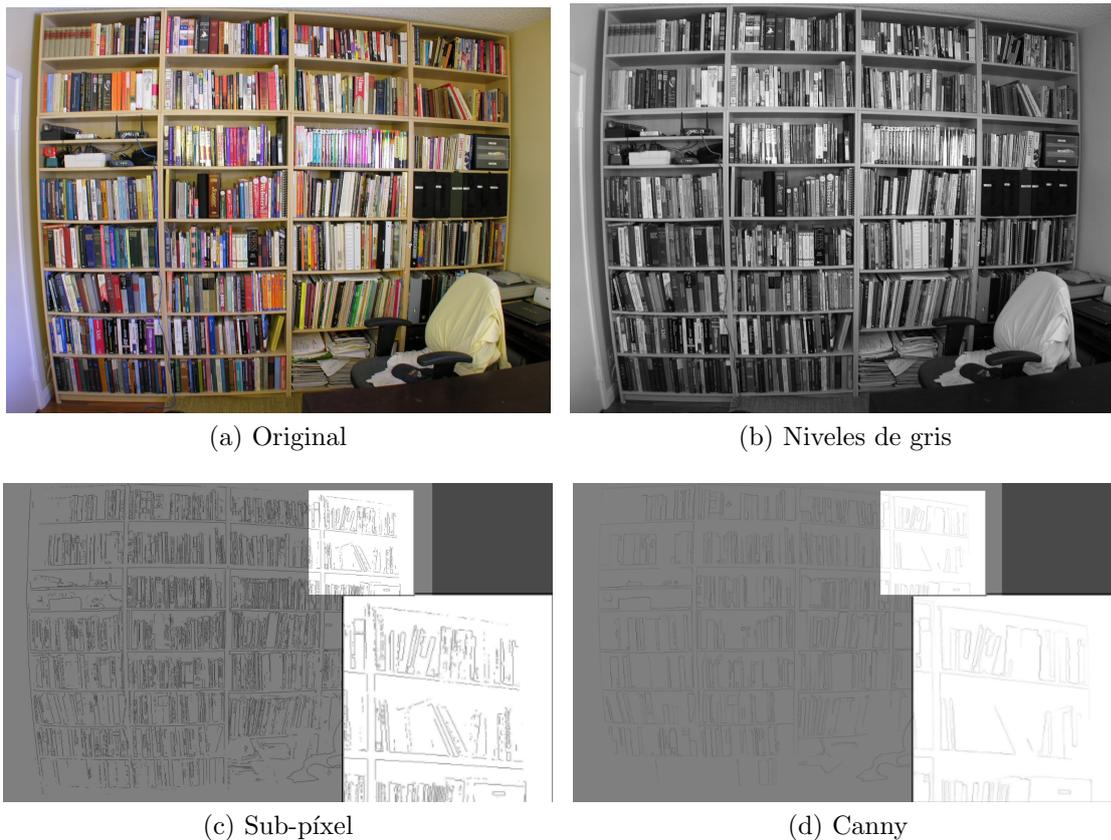


Figura 14: Comparación de detección usando los métodos descritos

Por tanto, ese valor lleva a considerar solamente como píxeles borde el 5 % de todos los píxeles.

En el caso del método de Canny se han usado  $k_{min} = 0,45$  y  $k_{max} = 0,98$  para el cálculo del umbral mínimo y máximo respectivamente. Al igual que en el caso anterior, para calcular dichos umbrales se parte de la imagen del modulo del gradiente. Los valores seleccionados harían que se consideraran como borde aquellos píxeles cuyo valor estuviera inscrito entre los umbrales obtenidos al calcular la mediana y dejar el 45 y el 98 % de los píxeles a la izquierda, respectivamente.

### 3.3. Corrección de la distorsión

Una vez desarrolladas las herramientas de detección de contornos, es la hora de usarlas para detectar las líneas y realizar la corrección de la distorsión de la imagen. En la figura 15 se puede ver cuál es el proceso completo llevado a cabo, incluyendo la detección de bordes.

A partir de la imagen de entrada se realiza una conversión a niveles de gris de la misma. Esta conversión se rige según la ecuación (24), que determina cuánto influye cada canal en el valor de gris para un píxel  $i$ .

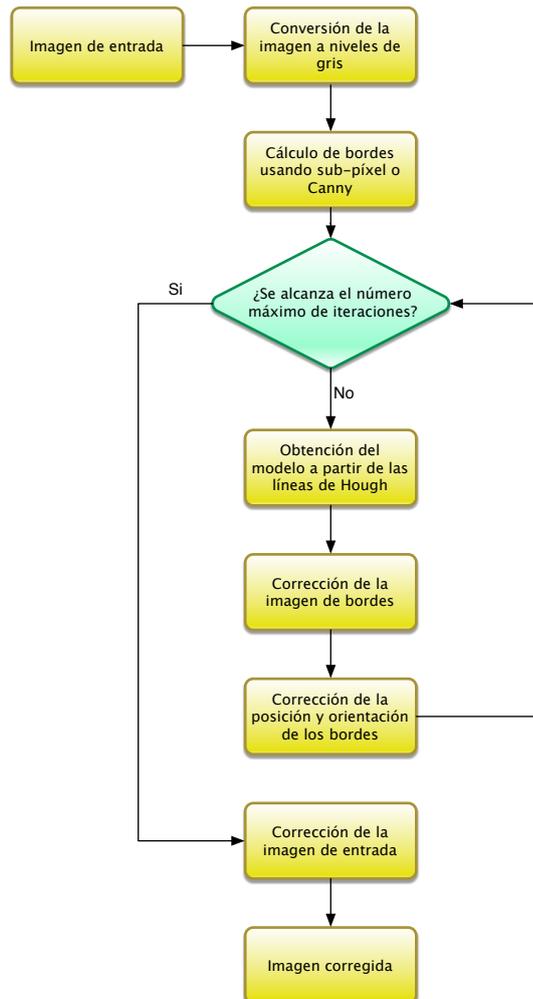


Figura 15: Diagrama de flujo de la corrección iterativa de la distorsión

$$gray_i = 0,3R_i + 0,59G_i + 0,11B_i \quad (24)$$

Una vez se tiene la imagen en niveles de gris, se pasa a la detección de contornos. Para ello se puede usar cualquiera de los dos métodos descritos con anterioridad. A partir de ese momento comienza el proceso iterativo de corrección incremental. Se obtiene el modelo de distorsión de la lente a partir de la función de Hough. Con dicho modelo, se realiza la corrección de la imagen de los bordes detectados. Como además de los bordes se tienen también las posiciones de los mismos, así como la orientación, éstos también deberán ser corregidos usando el modelo.

Este procedimiento se aplica hasta alcanzar el número máximo de iteraciones. Una vez se llega a él, se corrige la imagen de entrada usando el modelo final obtenido, teniendo así la imagen con la distorsión corregida.

---

## 4. Resultados

En la sección anterior se han descrito las diferentes herramientas desarrolladas, así como su comportamiento. Además, se ha hecho una pequeña comparativa entre el resultado que ofrecen los dos sistemas de detección de contornos implementados sobre una misma imagen.

En esta sección se empezará por hacer una comparación mas exhaustiva de los resultados obtenidos a partir de los dos diferentes detectores de bordes seleccionados. Además, se mostrarán los resultados de la corrección de la distorsión, tanto sobre imágenes sintéticas como en imágenes reales. Se consideran también como sintéticas aquellas en las que aparece el patrón de calibración, ya que se trata de una estructura conocida y artificial.

En cualquier caso, se mostrarán resultados usando tanto un detector de contornos como otro. Se indicarán además el número de iteraciones necesarias para la obtención del resultado. En cuanto al umbral usado para la detección de contornos, en el caso del detector sub-píxel se ha usado un valor de  $k = 0,95$ , mientras que en el método de Canny se tiene  $k_{min} = 0,8$  y  $k_{max} = 0,98$ . La técnica para la obtención de los umbrales a partir de estos valores ya fue descrita en la sección anterior.

### 4.1. Detectores de contornos

En estas sección se hará una comparación más en profundidad de los resultados obtenidos con los dos detectores de contornos implementados: el método sub-píxel y el método de Canny.

Para estudiar los diferentes resultados que se obtienen a partir de estas herramientas se variará el valor de  $k$ . Este valor representa en tanto por uno el porcentaje de valores que se dejan a la izquierda al hacer el cálculo de la mediana sobre los valores de la imagen del módulo del gradiente. Esto permite realizar un cálculo semiautomático del valor umbral necesario para discriminar cuándo se considera o no a un píxel como borde.

En el caso de las imágenes obtenidas con el método de Canny, este valor de  $k$  será doble, diferenciándose entre  $k_{min}$  y  $k_{max}$ . Estos valores están relacionados con el valor de umbral mínimo y máximo respectivamente. Al igual que cuando se calcula el umbral para el metodo sub-píxel, cada uno de estos valores de representa el porcentaje de píxeles a dejar a la izquierda con el cálculo de la mediana.

#### 4.1.1. Sub-píxel

En la figura 16 se muestran los resultados de aplicar el métodos sub-píxel variando el valor de  $k$  entre 0,7 y 0,95 con paso 0,05.

Como era de esperar, mientras menor es el valor de  $k$ , menor es el valor del umbral y, por tanto, aparecen mayor cantidad de bordes en la imagen. A medida que este valor va aumentando,

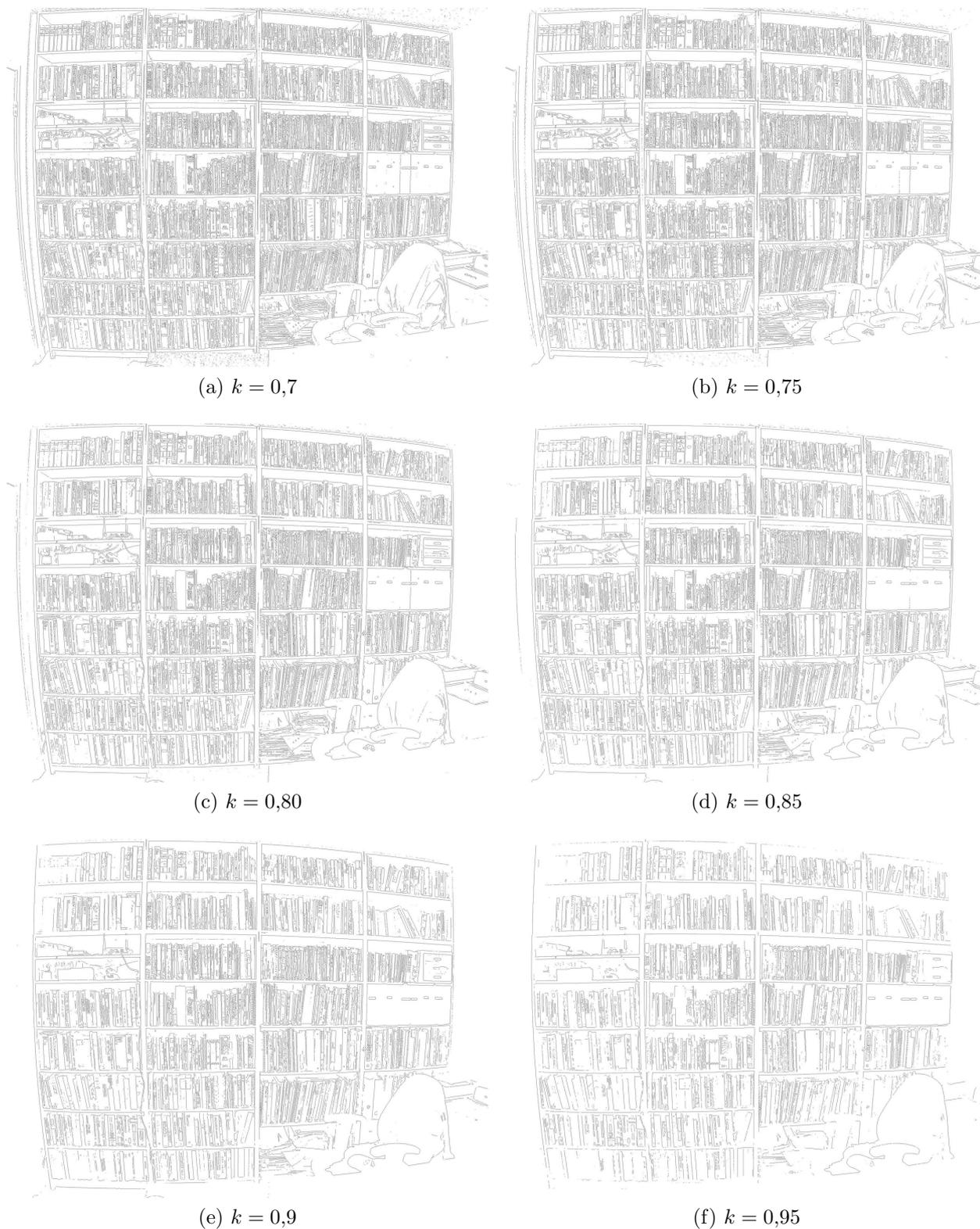


Figura 16: Resultado del método de detección de contornos con precisión sub-píxel para diferentes valores del porcentaje para el cálculo del umbral

consecuentemente aumenta el valor del umbral, haciendo de esta manera que la aparición de contornos en la imagen resultante sea inferior.

No obstante, a pesar de que se suba el valor del umbral, aparecen pequeños detalles en la imagen. Probablemente este resultado sea fruto de pequeñas perturbaciones o artefactos presentes en la imagen, al tratarse de una de una fotografía real.

#### 4.1.2. Canny

En la figura 15 se tiene el resultado de aplicar el método de Canny a la misma imagen que se usó para el método sub-píxel. Esto permite comparar en igualdad de condiciones los resultado con respecto a los anteriores. En relación a los valores de  $k$  usados para el cómputo de los umbrales, se ha mantenido el valor del umbral superior constante y se ha variado el umbral inferior.

Numéricamente, para el umbral superior se ha mantenido un valor de  $k_{max} = 0,98$ . Sin embargo, para el umbral inferior se tiene  $k_{min} \in [0,1 \dots 0,95]$  con paso 0,05. De esta manera se tienen un total de diecinueve resultados. Para valores altos de dicho umbral se puede ver que se pierden algunos resultados, pero no parece ser tan apreciable como en el caso del método sub-píxel.

Comparativamente, ambos detectores de bordes parecen dar buenos resultados. Sin embargo, en el caso del método de Canny, la imagen de contornos resultante es más limpia. Es decir, no presenta tantos pequeños detalles como el el caso del algoritmo sub-píxel, el cual conserva estos detalles inclusive cuando el valor de  $k$  para el cálculo del umbral es alto. La aparición de menos detalles probablemente esté justificada principalmente por dos aspectos. En primer lugar, por el operador DroG, que incorpora previamente al cálculo del módulo del gradiente la convolución con una gaussiana. Además, el proceso de supresión de no máximos asegura que el píxel borde que se considera es máximo en su orientación con respecto a su vecindad.

Además de ello, el método de Canny parece conservar una mayor continuidad respecto a los píxeles del contorno. Esto es debido al proceso de histéresis, en el cual se va explorando recursivamente la vecindad y estableciéndose cierta conectividad entre aquellos píxeles que cumplen la condición de ser superiores al umbral mínimo e inferiores al máximo.

En las siguientes secciones se observará si estas diferencias entre ambos métodos de detección de contornos proporcionan alguna diferencia a la hora de realizar el ajuste a rectas a través de la transformada de Hough y la posterior estimación del modelo de distorsión de la lente.



(a)  $k_{min} = 0,1$



(b)  $k_{min} = 0,15$



(c)  $k_{min} = 0,2$



(d)  $k_{min} = 0,25$



(e)  $k_{min} = 0,3$



(f)  $k_{min} = 0,35$

(g)  $k_{min} = 0,4$ (h)  $k_{min} = 0,45$ (i)  $k_{min} = 0,5$ (j)  $k_{min} = 0,55$ (k)  $k_{min} = 0,6$ (l)  $k_{min} = 0,65$



(m)  $k_{min} = 0,7$



(n)  $k_{min} = 0,75$



(ñ)  $k_{min} = 0,8$



(o)  $k_{min} = 0,85$



(p)  $k_{min} = 0,9$



(q)  $k_{min} = 0,95$

Figura 15: Resultado del método de detección de contornos de Canny para diferentes valores del porcentaje para el cálculo del umbral inferior

## 4.2. Imágenes sintéticas

En la figura 16 se puede ver el resultado de aplicar el método a un patrón de calibración sintético que ha sido distorsionado de manera artificial.

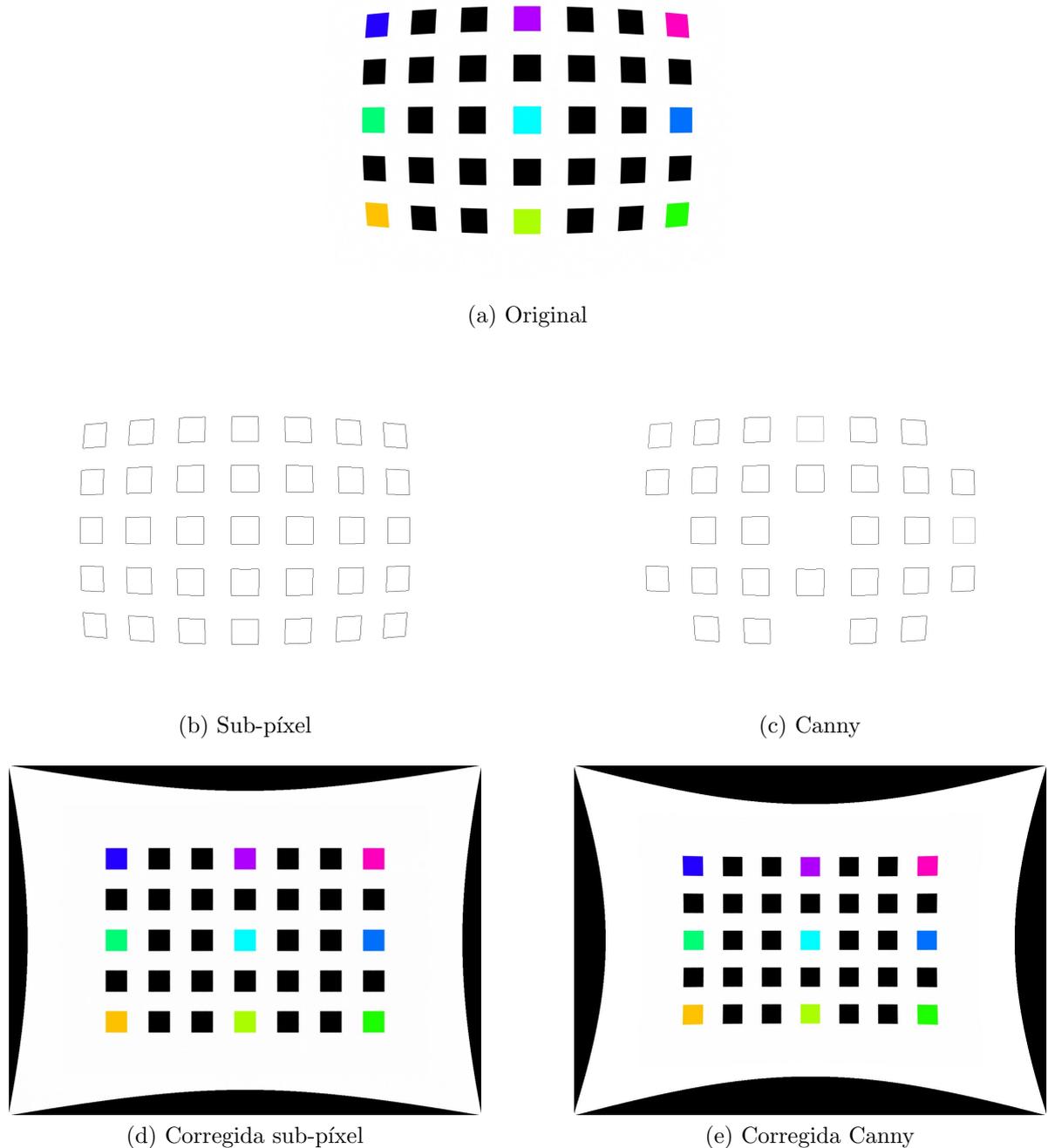
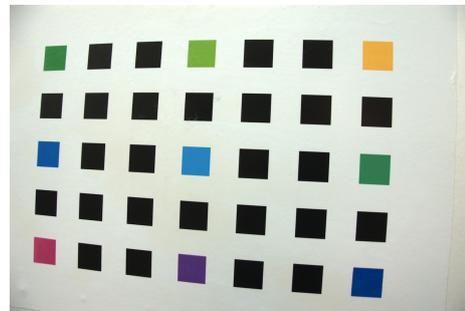


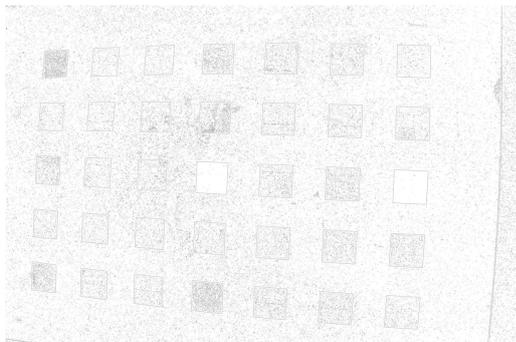
Figura 16: Corrección de la distorsión en una imagen sintética distorsionada artificialmente

La primera diferencia que puede observarse es que la detección de contornos realizada mediante el método de Canny parece no haber detectado todos los cuadrados del patrón. Esto

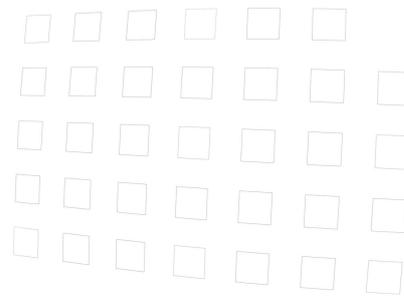
posiblemente sea debido a que el valor para el umbral mínimo considerado para el proceso de histéresis sea un poco alto. No obstante la primera cosa a destacar es el número de iteraciones necesario para obtener el resultado. Mientras que en el caso del uso del método de detección sub-píxel el número de iteraciones para la corrección ha sido de cinco, al usar el método de Canny sólo ha sido necesaria una iteración.



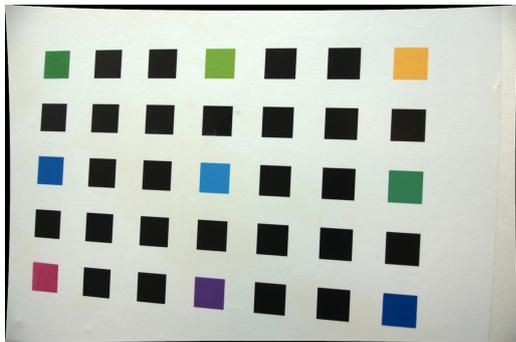
(a) Original



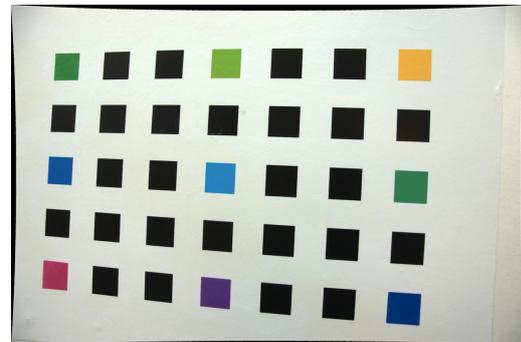
(b) Sub-píxel



(c) Canny



(d) Corregida sub-píxel



(e) Corregida Canny

Figura 17: Corrección de la distorsión en una fotografía del patrón de calibración

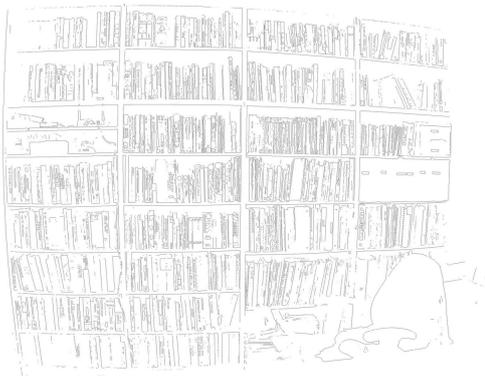
En la figura 17 se puede ver el resultado de aplicar el método a una fotografía de un patrón de calibración. Al igual que antes se muestra la imagen de entrada, las detecciones de contornos y las imágenes corregidas. En este caso es muchísimo más apreciable que el resultado de aplicar la detección de bordes por Canny proporciona un resultado mucho más limpio que en el caso del uso del método sub-píxel. Nuevamente, si se comparan el número de iteraciones necesarias para obtener el resultado, en el caso de usar el método sub-píxel han sido necesarias cinco iteraciones, mientras que en el caso de Canny sólo se ha sido necesario usar dos.

### 4.3. Imágenes reales

En el caso de imágenes reales se van a mostrar tres resultados: la imagen de una estantería<sup>2</sup>, la de una estación de metro<sup>3</sup> y la de un cuadro<sup>4</sup>.



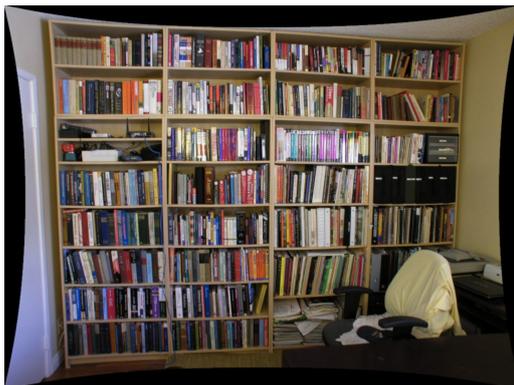
(a) Original



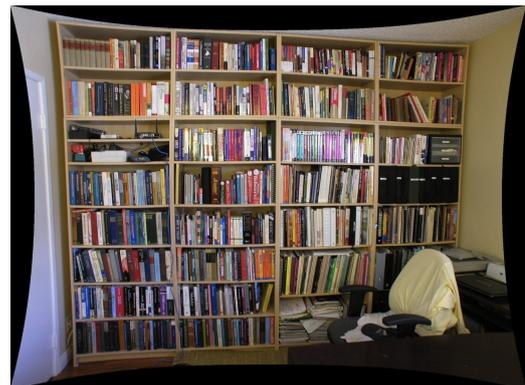
(b) Sub-píxel



(c) Canny



(d) Corregida sub-píxel



(e) Corregida Canny

Figura 18: Corrección de la distorsión en una fotografía de una estantería

<sup>2</sup>Imagen cortesía de Jon Delorey. <http://www.flickr.com/photos/jonsphotos/200626160/> Licencia CC-BY-NC-SA

<sup>3</sup>Imagen cortesía de silk cut. <http://www.flickr.com/photos/silkcut/3186267799/> Licencia CC-BY-NC

<sup>4</sup>Imagen cortesía de litlnemo. <http://www.flickr.com/photos/litlnemo/6918940761/> Licencia CC-BY-NC-SA

De la misma forma que se hiciese en el apartado anterior, se mostrará la imagen de entrada, así como la comparativa entre la detección de contornos sub-píxel y Canny, y la imagen corregida resultante.

En la figura 18 se puede ver el resultado de aplicar el proceso a la imagen de una estantería.

De la misma forma que sucediera que en caso anterior de la fotografía del patrón de calibración, el resultado de la detección de contornos por medio del método de Canny es mucho más limpia que la realizada usando el método sub-píxel. Para la realización de la corrección, en el caso del sub-píxel se han necesitado un total de diez iteraciones mientras que en el método de Canny sólo ha sido necesaria una.

En la figura 19 se muestran los resultados de aplicar el proceso a una imagen de una estación de metro.

Nuevamente se aprecia una mayor limpieza en la detección de contornos por medio del método de Canny. Para realizar la corrección de la imagen de entrada, en el método sub-píxel han sido necesarias un total de cinco iteraciones, mientras que para el de Canny se han realizado dos.

En cuanto a los resultados sobre la fotografía de un cuadro, éstos pueden observarse en la figura 20.

Los resultados obtenidos son semejantes a los anteriores. Al igual que en el caso anterior, para el método sub-píxel se han hecho cinco iteraciones de corrección, mientras que para el de Canny se han hecho dos.

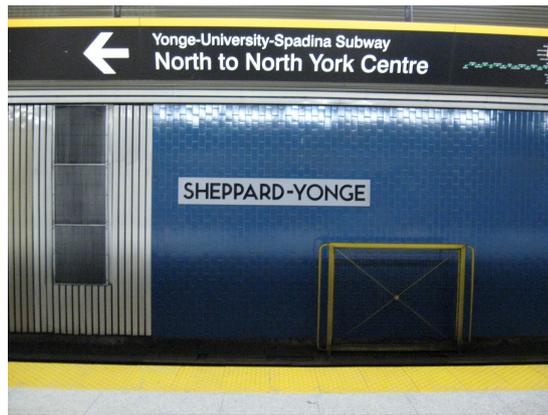
### 4.4. Líneas de Hough en función del detector de bordes

Hasta ahora se han mostrado los diferentes resultados tanto para imágenes sintéticas (las del patrón de calibración), como para imágenes reales. Una de las principales diferencias que se ha observado es que cuando se hace uso del método de Canny en el paso de detección de contornos, previo al bucle de detección de líneas y corrección, el resultado se obtiene en un número inferior de iteraciones.

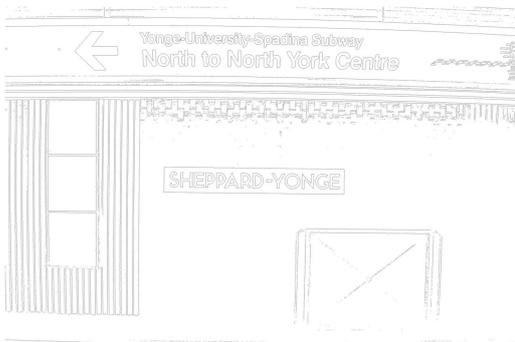
Teniendo esto en cuenta, en esta sección se va a estudiar qué es lo que está sucediendo a la hora de detectar las líneas y cuál es la razón por la que el método de Canny hace que se estime mejor el modelo de distorsión de la lente que cuando se usa el método de detección de contornos sub-píxel.

Para ilustrarlo se usará la última imagen de la sección anterior de resultados sobre imágenes reales, la fotografía de un cuadro.

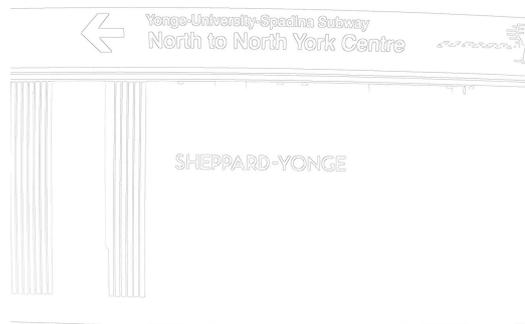
En la figura 20 se puede ver la evolución de la detección de las líneas mediante la transformada de Hough, a lo largo de las diferentes iteraciones. Para este resultado se ha partido de una detección de contornos mediante el método sub-píxel. En las primeras iteraciones se



(a) Original



(b) Sub-píxel



(c) Canny



(d) Corregida sub-píxel



(e) Corregida Canny

Figura 19: Corrección de la distorsión en una fotografía de una estación de metro

puede ver que en la detección de las líneas aparecen puntos que pertenecen a una misma línea de diferentes colores. Esto es así porque un punto recibe un color en función a la línea a la que pertenece. Inicialmente, es imposible aproximar las líneas distorsionadas a una recta. Por ello en las primeras iteraciones, líneas que presumiblemente son las mismas, aparecen con puntos de diferentes colores.

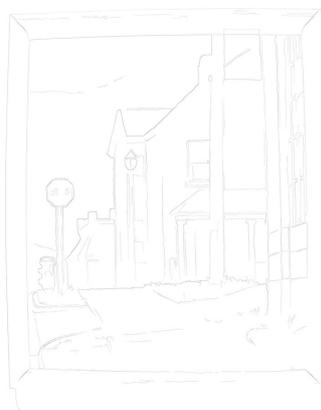
A medida que progresan las iteraciones se puede ver una corrección progresiva de la dis-



(a) Original



(b) Sub-píxel



(c) Canny



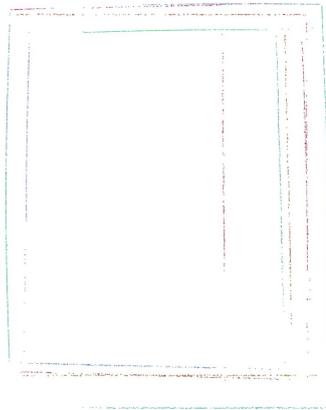
(d) Corregida sub-píxel



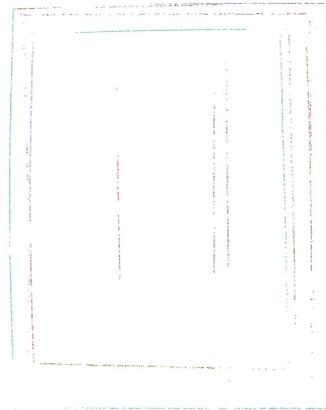
(e) Corregida Canny

Figura 20: Corrección de la distorsión en una fotografía de un cuadro

torsión. Este hecho es patente en la tendencia de las líneas detectadas a estar más rectas que



(a) Iteración 1



(b) Iteración 2



(c) Iteración 3



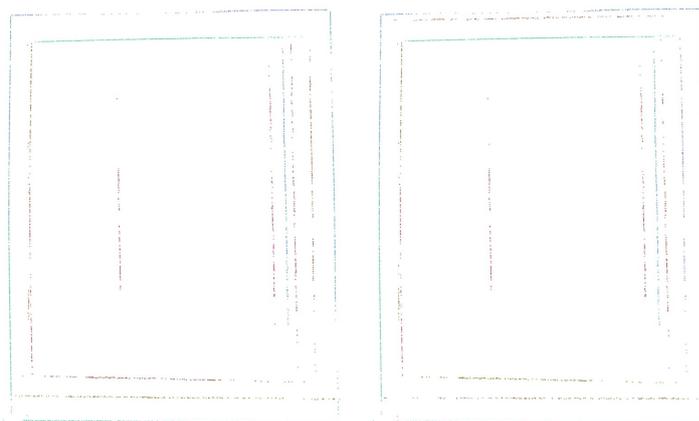
(d) Iteración 4



(e) Iteración 5

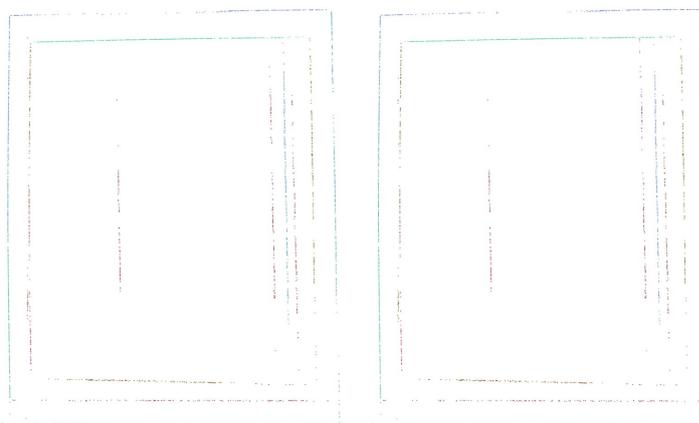


(f) Iteración 6



(g) Iteración 7

(h) Iteración 8



(i) Iteración 9

(j) Iteración 10

Figura 20: Evolución de las líneas detectadas por la transformada de Hough a través de las diferentes iteraciones, partiendo de una detección de contornos por medio del método de detección sub-píxel

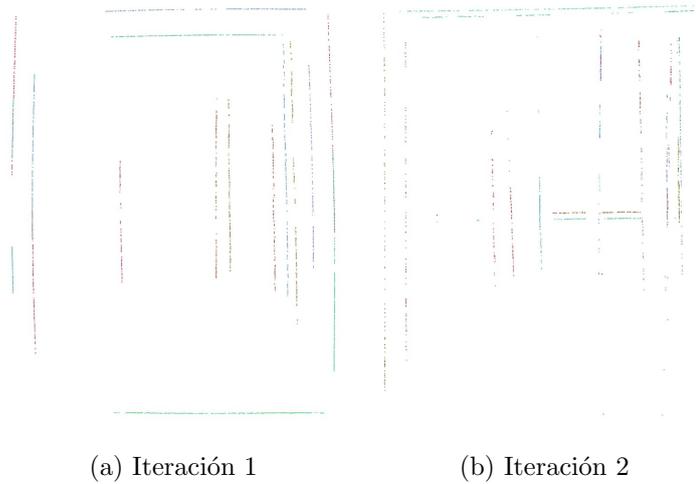


Figura 21: Evolución de las líneas detectadas por la transformada de Hough a través de las diferentes iteraciones, partiendo de una detección de contornos por medio del método de Canny

en las primeras detecciones observadas. No obstante, aunque parece que se obtiene una buena corrección, en detalle parece que aún las rectas detectadas mezclan algunos colores. Por tanto, esos puntos no pertenecen a una misma recta como cabría esperar.

En la figura 21 puede verse la evolución de las líneas detectadas por la transformada de Hough partiendo de la detección de contornos realizada con el método de Canny. En este caso sólo se muestran dos iteraciones, que como se dijo en la sección anterior, fueron suficientes para la corrección. Puede verse como ya en la segunda de ellas las líneas detectadas aparecen rectas. Al contrario de lo que sucedía con el método anterior, aparecen líneas completamente de un mismo color. Es más, estas líneas son las más largas de la imagen, lo cual permite una mejor estimación del modelo de distorsión de la lente.

Esto hecho viene justificado porque el método de Canny proporciona líneas con un mayor número de puntos, permitiendo que a la transformada de Hough le sea más sencillo detectar líneas más largas. Por esta razón, para obtener el resultado esperado, hacen falta un número inferior de iteraciones que en el caso en el que se hace uso del método de detección de contornos con precisión sub-píxel.

## 5. Conclusiones y Líneas Futuras

A lo largo del presente trabajo se han descrito diferentes métodos, tanto para la detección de puntos de interés dentro de una imagen como para la agrupación de estos puntos en primitivas. Como herramientas de detección de puntos de interés se ha hablado de dos detectores de contornos: el método sub-píxel y el método de Canny. El método sub-píxel se describe como un método que considera al borde como una discontinuidad en los valores de intensidad de la imagen, al contrario que los métodos tradicionales. Esta técnica proporciona la posición sub-píxel, orientación y valores de intensidad a ambos lados del contorno. Por otra parte, el método de Canny representa la búsqueda de un contorno basándose en tres principios fundamentales: buena detección, buena localización y respuesta única. Para ello usa el operador  $DroG$ , la supresión de no máximos y el proceso de histéresis.

Como método de agrupación de los puntos en líneas rectas se ha hablado del método de Hough. Para ello se transforma el espacio a un espacio transformado de Hough, en el que las rectas vienen representadas por la distancia al origen y su orientación. Cada punto contribuye a la recta en función de esos parámetros e incrementa la posición en un acumulador de votaciones. Se han propuesto dos variaciones a este método. Por un lado se ve la inclusión de una variación del peso de votación en función de la distancia, haciendo que contribuyan más a la recta aquellos píxeles que se encuentran más cercanos a la misma. Además, también se propone una corrección en la orientación tangente al contorno, que permite acotar mejor el rango angular de búsqueda de las rectas más cercanas.

La unión de las técnicas de detección con la agrupación en líneas ha permitido obtener un método de estimación del modelo de distorsión que una lente produce sobre una imagen. Considerando los resultados obtenidos, esta estimación parece ser más exacta en el caso del uso del método de Canny en lugar que el sub-píxel para la detección de los contornos. Esto es debido a que los procesos de supresión de no máximos y de histéresis proporcionan bordes más limpios y con mayor número de puntos. Este hecho provoca que se encuentren rectas más largas con mayor número de puntos asociados. De esta manera se proporciona una mayor información a la estimación del modelo y queda reflejado en la necesidad de un número inferior de iteraciones en el proceso de corrección.

El método propuesto consigue automatizar el proceso de detección de primitivas, en lugar de hacer que éstas sean indicadas de manera manual. Ello unido a un proceso iterativo incremental de corrección, ha permitido corregir la distorsión presente en las imágenes de entrada, tanto sintéticas como reales.

Si bien, existen aspectos que pueden mejorarse de cara al futuro. El procesamiento tanto de los contornos como su agrupación en líneas rectas son susceptibles de paralelización. En ese sentido se propone el uso de librerías como OpenMP, que permite adaptarse de forma automática al número de procesadores presentes en el equipo que se use. Esto permitiría acelerar partes del código, haciendo que la ejecución fuese más rápida. Otra posibilidad es el uso de la tecnología presente en las GPU, como por ejemplo la librería CUDA de NVidia, que posibilita explotar el paralelismo de todos los procesadores presentes en las tarjetas gráficas.

---

El método propuesto es dependiente del número de iteraciones de corrección. Al poner un número de iteraciones superior, además de ejecutarse de forma más lenta, se corre el riesgo de deformar la imagen de entrada en lugar de simplemente corregir la distorsión de la lente. Sería posible estudiar la estimación del modelo de distorsión óptimo, considerando aquél en el cuál se ajusten un mayor número de rectas al conjunto de puntos obtenidos a partir de la detección de contornos. Esto haría independiente el número de iteraciones respecto del modelo de distorsión obtenido, puesto que en cualquier caso el sistema se quedaría con el óptimo.

Anteriormente se proponía como mejora futura la paralelización del código. Este tipo de procesamiento adquiere una importancia superior cuando en lugar de procesar una imagen aislada, se pasa a procesar secuencias de vídeo. La rapidez en esos casos es vital, puesto que se entra en sistemas denominados de tiempo real. En este sentido, no sólo sería importante la paralelización, si no también estudiar si sería posible aprovechar información de *frames* anteriores, que permitan acelerar el proceso en el que se vaya a procesar.

Por último, al principio se hablaba de que en el modelo proyectivo simple un objeto 3D de la escena real se proyecta en 2D a través de coordenadas proyectivas. También es posible proyectar objetos virtuales 3D de tal manera que parece como si estuvieran en la escena real. Esta técnica se denomina realidad aumentada. En aplicaciones de este tipo sería interesante tener un modelo fiable de la distorsión de la lente, de tal manera que la proyección del objeto virtual tuviese en cuenta dicha distorsión y conseguir que la inclusión del objeto virtual fuese lo más realista posible.

## Referencias

- [1] Luis Alvarez, Luis Gómez, and Pedro Henríquez. Zoom dependent lens distortion mathematical models. *Journal of Mathematical Imaging and Vision*, (44):480–490, 2012.
- [2] Luis Alvarez, Luis Gómez, and J. Rafael Sendra. An algebraic approach to lens distortion by line rectification. *Journal of Mathematical Imaging and Vision*, (35):36–50, 2009.
- [3] Luis Alvarez, Luis Gomez, and J. Rafael Sendra. Algebraic lens distortion model estimation. *Image Processing On Line*, 2010.
- [4] Luis Alvarez, Luis Gómez, and J. Rafael Sendra. Accurate depth dependent lens distortion models: An application to planar view scenarios. *Journal of Mathematical Imaging and Vision*, (39):75–85, 2011.
- [5] John Canny. A computacional approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 8(6):679–698, nov. 1986.
- [6] Leandro A.F. Fernandes and Manuel M. Oliveira. Real-time line detection through an improved Hough transform voting scheme. *Pattern Recognition*, 41:299–314, 2007.
- [7] Rafael C. González and Richard E. Woods. *Tratamiento Digital de Imágenes*. Addison - Wesley, 1996.
- [8] Javier González Jiménez. *Visión por Computador*. Paraninfo, 2000.
- [9] Karl Krissian and Santiago Aja-Fernández. Noise-driven anisotropic diffusion filtering of mri. *IEE Transactions on Image Processing*, 18(10):2265–2274, 2009.
- [10] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.
- [11] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson, 2008.
- [12] A. Trujillo-Pino, K. Krissian, D. Santana-Cedrés, J. Esclarín-Monreal, and J.M. Carreira-Villamor. A subpixel edge detector applied to aortic dissection detection. In Roberto Moreno-Díaz, Franz Pichler, and Alexis Quesada-Arencia, editors, *Computer Aided Systems Theory - EUROCAST 2011*. Springer Berlin Heidelberg, February 2012.
- [13] Agustín Trujillo Pino. *Localización de contornos con precisión sub-píxel en imágenes bidimensionales y tridimensionales*. PhD thesis, Universidad de Las Palmas de Gran Canaria, 2004.
- [14] Agustín Trujillo-Pino, Karl Krissian, Miguel Alemán-Flores, and Daniel Santana-Cedrés. Accurate subpixel edge location based on partial area effect. *Image and Vision Computing*, page 19, 2012.