

Detección Automática de un Patrón para la Calibración de Cámaras

Arturo de la Escalera*, Jose María Armingol*, Jose Luis Pech**, José Julián Gómez**

*Grupo de Sistemas Inteligentes, Universidad Carlos III de Madrid, Avda de la Universidad 30, 28911, Leganés, Madrid, España (e-mail: escalera@ing.uc3m.es, armingol@ing.uc3m.es)

**Solex Vision Artificial S.L. Avda. de la Astronomía 43, 28830 San Fernando de Henares, Madrid, España (jose Luis@solexvision.com, josejulian@solexvision.com)

Resumen: Cada vez se requieren más aplicaciones en las que es necesaria la calibración de una cámara para poder realizar mediciones sobre las imágenes. En la actualidad se dispone de una serie de algoritmos capaces de obtener estos valores de forma semi-automática, por lo que se están investigando métodos para calcularlos de la manera más automática posible ahorrando un gran tiempo a los usuarios. El método que se propone en este artículo utiliza un patrón similar a un tablero de ajedrez, que es encontrado en cada imagen de forma automática sin información previa del número de filas y columnas. Los conjuntos de líneas que forman el patrón son encontrados mediante un análisis conjunto de la transformada de Hough, esquinas e invariantes a la transformación de la perspectiva. Se muestran varios ejemplos y su comparación con otros métodos más tradicionales. Copyright © 2010 CEA.

Palabras Clave: Análisis de Errores, Calibración de cámaras, Distorsión de Imágenes, Reconocimiento de Patrones, Visión por Computador

1. INTRODUCCIÓN

A la hora de diseñar un sistema de Visión por Computador siempre se tienen en cuenta una serie de parámetros que serán decisivos para que la apariencia de los objetos en la imagen sea la mejor de cara a los posteriores algoritmos de análisis. Así se elegirá una óptica con una distancia focal, f , que permita que se observe con el suficiente tamaño en la imagen el elemento a buscar o que tenga una lente con la suficiente calidad como para que los objetos no se vean deformados. Respecto a la colocación de la cámara, ésta se realizará de forma que se perciba de la mejor manera posible el espacio por el que pueden aparecer los objetos. Para muchas aplicaciones estas medidas son suficientes. Sin embargo, para aquellos algoritmos de Visión por Computador que necesiten extraer información 3D de una imagen o una secuencia de ellas o establecer la correspondencia entre dos o más cámaras, la calibración de los parámetros intrínsecos y extrínsecos del sistema de visión es una etapa fundamental.

El caso concreto que ha motivado este estudio es el proyecto *VISIÓN: Comunicaciones de Vídeo de Nueva Generación* (Vision 2007), en el que se dispone de más de 40 cámaras (Figura 1) que se dedicarán a: reconstrucción estéreo (18 cámaras), *Visual Hull* (18 cámaras), y seguimiento de objetos (entre 4 y 6 cámaras). Estas tres tareas necesitan eliminar la distorsión de cada cámara, encontrar correspondencias entre imágenes provenientes de cámaras distintas, y realizar mediciones 3D. Por ello, tal como se detalla más adelante, no sirve conocer los valores nominales de cada cámara sino que hay que caracterizarla y saber su posición geométrica respecto a las demás y a un sistema externo de referencia. Además el alto número de equipos hace recomendable que el proceso de calibración sea rápido. Por último no se puede pedir que este proceso sea realizado siempre por personal especializado por lo que debe de ser lo más automático posible.



Figura 1. Distribución de las cámaras en el proyecto VISION (www.cenit-vision.org)

Los valores a obtener se pueden clasificar en dos grupos:

- **Parámetros Intrínsecos:** hacen referencia al conjunto cámara-óptica y son las distancias focales horizontal, f_x , y vertical, f_y , las coordenadas del centro óptico (u_0, v_0), y las distorsiones de la lente: radial, k , y longitudinal, p .
- **Parámetros Extrínsecos:** hacen referencia a la colocación física de la cámara que viene definida por la posición y orientación de ésta respecto al sistema global de coordenadas del mundo. Se toma el criterio de que primero ha habido una traslación de la cámara, \mathbf{T} , seguida de una rotación, \mathbf{R} .

Los parámetros extrínsecos y la proyección en la imagen, \mathbf{U} , de un punto en el espacio, \mathbf{X} , se pueden representar mediante la expresión matricial:

$$\mathbf{U} = \mathbf{P}\mathbf{X} = \mathbf{K}(\mathbf{R}|\mathbf{T})\mathbf{X}$$

$$\begin{pmatrix} u \\ v \\ s \end{pmatrix} = \begin{pmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (1)$$

Respecto a la distorsión, se sigue la fórmula:

$$\begin{aligned} x_r &= x_i + x_i(k_1 r^2 + k_2 r^4) + 2p_1 x_i y_i + p_2(r^2 + 2x_i^2) \\ y_r &= y_i + y_i(k_1 r^2 + k_2 r^4) + p_1(r^2 + 2y_i^2) + 2p_2 x_i y_i \end{aligned} \quad (2)$$

Siendo (x_r, y_r) las coordenadas reales de los píxeles, (x_i, y_i) las coordenadas ideales y r la distancia al centro óptico de la imagen.

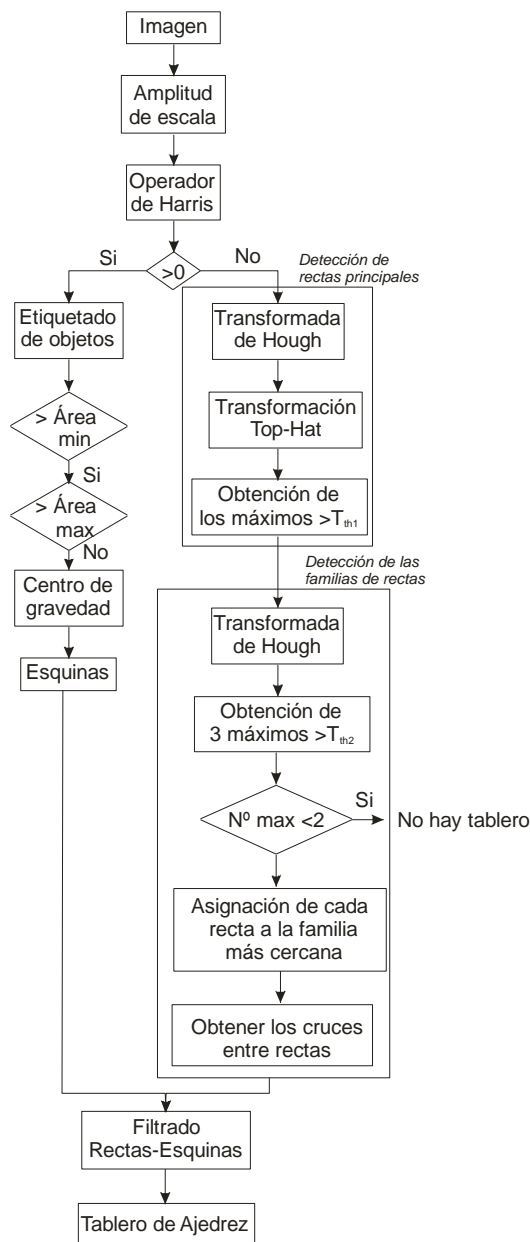


Figura 2. Etapas del método propuesto

La estructura del artículo es la siguiente. En el apartado 2 se introducen las ventajas de disponer de un sistema de calibración automático, también se hace una exposición del estado del arte sobre detección automática de puntos de calibración. Los siguientes apartados se centran en el algoritmo propuesto, cuyo conjunto de etapas pueden verse representadas en la figura 2. Así, en el apartado 3 se explicará cómo se obtienen de forma separada las dos características que definen al patrón. Para obtenerlas se aplica el operador de Harris que permite determinar las esquinas y es a su vez la entrada a una transformada de Hough que detecta las rectas de la imagen. De entre todas ellas habrá dos grupos formados por las rectas del patrón según se explica en el apartado 4. Los máximos de la primera transformada de Hough son los puntos de entrada de una

segunda transformada, en la que están representadas los haces convergentes del patrón de calibración. Cómo no todas las rectas convergentes tienen que pertenecer a éste, se contrasta la intersección de las rectas con las esquinas detectadas con anterioridad y se eliminan todas las rectas que no sean coherentes con el resto del conjunto hasta detectar el patrón. En el apartado 5 se muestra como el algoritmo facilita la tarea de obtención de los parámetros intrínsecos y extrínsecos de la cámara. Una serie de resultados y comparaciones con otros algoritmos que avalan la propuesta realizada se exponen en el apartado 6.

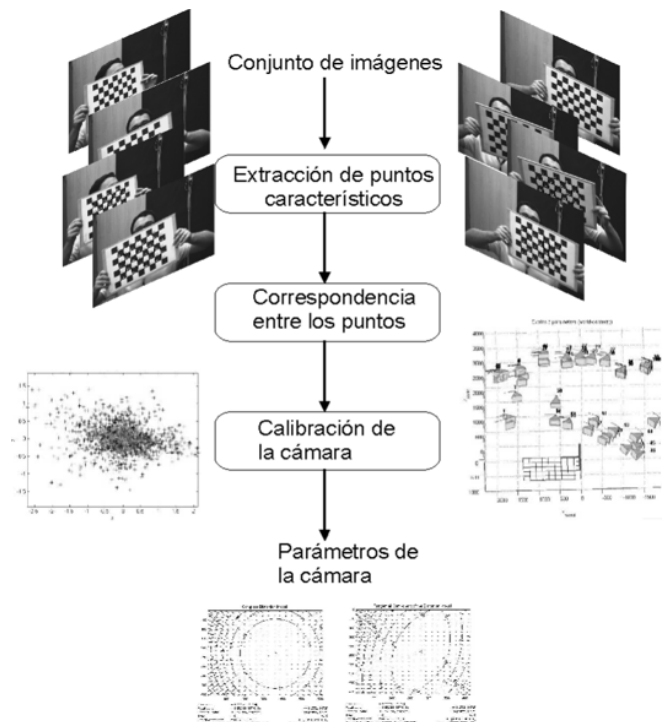


Figura 3. Etapas de la calibración de cámaras

2. ESTADO DEL ARTE

Los métodos de calibración (Figura 3) consisten generalmente en analizar en las imágenes la proyección de una serie de puntos característicos que se han situado en un objeto y cuyas coordenadas 3D son conocidas con gran precisión. Para ello se dispone de una secuencia de imágenes en las que se ha captado al patrón de calibración. En él, se encuentran una serie de puntos muy característicos de los que se conoce su apariencia y la distancia que hay entre ellos. Una vez que estos puntos han sido extraídos de la secuencia de imágenes y sus coordenadas 2D de las imágenes conocidas, se encuentra la correspondencia entre cada punto del patrón con toda la serie encontrada en la secuencia. Ésta es la entrada al algoritmo de calibración que dará los parámetros intrínsecos y extrínsecos de la cámara.

A lo largo de los años se han utilizado varios patrones de calibración: se comenzó con patrones tridimensionales (Trucco and Verri 1998) y aunque se han desarrollado algoritmos que detectan patrones unidimensionales como varios puntos situados en la misma línea (Zhang 2004) (Ronda *et al.* 2008), o las líneas de fuga (Zhaoxue and Pengfei 2004) (Grammatikopoulos *et al.* 2006), los más utilizados están formado por patrones bidimensionales (Zang 2000). Esto es debido a que los patrones de calibración planos son baratos y fáciles de construir dadas las

precisiones que las impresoras láseres tienen y la exactitud que demandan la mayoría de las aplicaciones de análisis de imágenes. Los primeros patrones estaban formados por matrices de círculos o cuadrados negros sobre fondo blanco ya que eran sencillos de umbralizar y obtener su centro de gravedad. Sin embargo se han ido abandonando ya que el centro de gravedad de las proyecciones en la imagen no coincide con el centro del objeto debido a la deformación del patrón que introduce la perspectiva. Es por ello que ahora se suelen utilizar patrones con una apariencia similar a los tableros de ajedrez ya que la alternancia de zonas claras y oscuras favorece la detección de las esquinas de los elementos del tablero.

El método de calibración antes referenciado (Zang 2000) es uno de los más populares debido tanto a su precisión como a que más tarde se implementó en un *toolbox* para *Matlab* (Bouguet) y en las librerías *OpenCv* desarrolladas por *Intel* (Bradski and Kaehler 2008). El método parte de varias imágenes en las que se detecta el patrón y se han hallado las coordenadas de cada una de las esquinas y se basa en el cálculo de homografías entre el patrón bidimensional y su proyección en la imagen. La implementación en *Matlab* pide al usuario que defina el número de columnas y filas del patrón así como que restrinja el área de búsqueda de las esquinas en la imagen. El algoritmo de las *OpenCv* parte de que el usuario ya ha realizado la correspondencia entre puntos.

La mayor parte de los trabajos de investigación sobre el tema de calibración se han centrado en la determinación de los parámetros mediante algoritmos robustos, pero han dejado de lado la tarea de determinar los puntos característicos que se utilizan como entrada de estos algoritmos. Se ha dejado al usuario la tarea de determinar las coordenadas de los puntos en las imágenes al ser un trabajo mecánico y rutinario.

Este hecho, que se podía justificar cuando no se disponía de buenos algoritmos de calibración, tiene una serie de desventajas en la actualidad:

- Al ser un proceso mecánico es fácil que el usuario deje de prestar la suficiente atención y por tanto está sujeto a errores, difíciles de detectar.
- Es creciente el número de aplicaciones y campos de investigación que utilizan un número elevado de cámaras, por lo que la calibración utilizando un número alto de imágenes debería realizarse todo lo automáticamente que sea posible.
- Aunque la calibración de los parámetros intrínsecos de las cámaras se puede realizar solamente una vez, no ocurre lo mismo con los extrínsecos que pueden cambiar varias veces hasta que se alcanza la configuración definitiva.
- Todos estos pasos se están realizando hasta ahora por personal formado que sabe elegir las posiciones adecuadas de los patrones pero dicha situación debe cambiar si se quiere que estas aplicaciones sean asumidas por usuarios sin una formación específica en Visión por Computador.

Por ello, se está investigando en la determinación automática de los puntos de un patrón de calibración que aparezca en una secuencia de imágenes desde diversas posiciones.

Una opción es que cada elemento del patrón tenga un código único que ayude a su detección y clasificación. Así (Fiala and Shu 2008) utilizan en lugar de cuadrados etiquetas especiales que se auto identifican similares a las Datamatrix (Figura 4.a).

En (Ahn et al. 2001) se utiliza un elemento circular que tiene un código de puntos interno (Figura 4.b). Otro patrón circular se usa en (Forbes et al. 2002) solo que ahora es la corona circular del patrón la que está codificada para diferenciarse entre ellos (Figura 4-c).

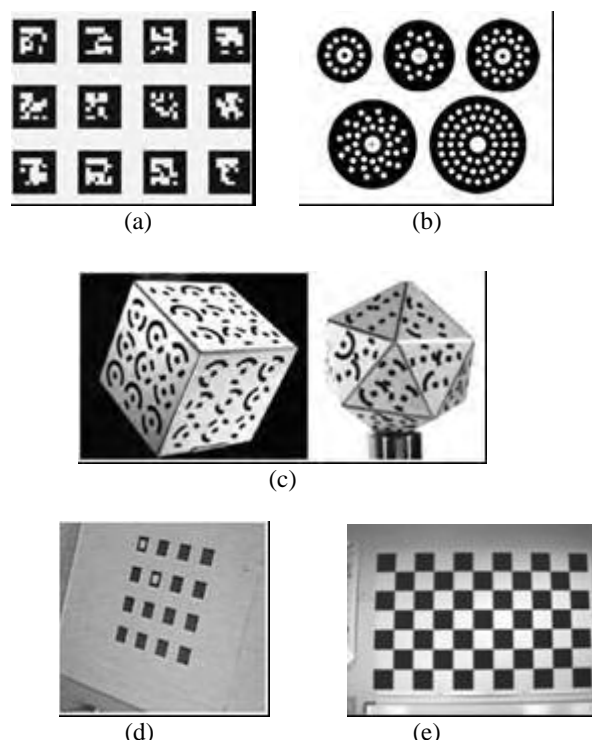


Figura 4. Varios patrones de calibración bidimensionales (a) (Fiala and Shu 2008) (b) (Ahn et al. 2001) (c) (Forbes et al. 2002) (d) (Matas et al.1997) (e) Tablero de ajedrez (Zang 2000)

Dentro del grupo de algoritmos en el que todos los elementos son similares se tiene el trabajo de (Matas et al. 1997) en el que utilizan el patrón diseñado por (Tsai 1987) consistente en una matriz de cuadrados que no se tocan (Figura 4-d). Dos de ellos tienen un cuadrado interior claro para determinar la esquina superior izquierda. Se detectan los bordes, que se agrupan en cuadrados. Después de desechar posibles candidatos por no ser lo suficientemente compactos y convexos, utilizan *graph matching* para detectar el patrón.

El resto de métodos usan el tablero de ajedrez presentado por (Zhang 2000) (Figura 4-e). En (Shu et al. 2003) utilizan un tablero de ajedrez al que han añadido tres círculos para dotarlo de la orientación. Primero, encuentran las esquinas con el detector de Harris (Harris and Stephens 1988) y las asocian mediante la triangulación de Delaunay ya que así se asegura que los triángulos son formados siempre por los puntos que están más próximos. Estos son agrupados en cuadriláteros y después en filas y columnas. El método falla cuando el patrón está demasiado oblicuo respecto a la cámara y existe mucha perspectiva. En (Yu and Peng 2006) han colocado cinco patrones formados por dos triángulos unidos por el vértice que serán buscados automáticamente como alternativa a que sea un usuario el que determine la zona donde se encuentra el tablero. Para ello utilizan correlación normalizada con modelos bajo diversas perspectivas. (Wang et al. 2007) utilizan un tablero de ajedrez en el que detecta las esquinas de la imagen. Determinan cuáles son debidas a la intersección entre esquinas blancas y negras. Para

terminar de diferenciarlas con las del resto de los objetos tiene que encontrar los dos grupos de líneas que corresponden a las filas y las columnas del tablero. Para ello detecta los dos puntos de fuga más importantes en la imagen. En (Douskos et al. 2007) también comienzan con la detección de las esquinas. Tras determinar el punto que tiene como coordenadas la mediana de las anteriores comienza a construir el tablero. Para ello ordenan los puntos alrededor del primero y calculan las direcciones principales analizando el gradiente. En (Douskos et al. 2008) perfeccionan el algoritmo añadiendo los puntos de fuga y la retro-proyección de los resultados para afinar los cálculos.

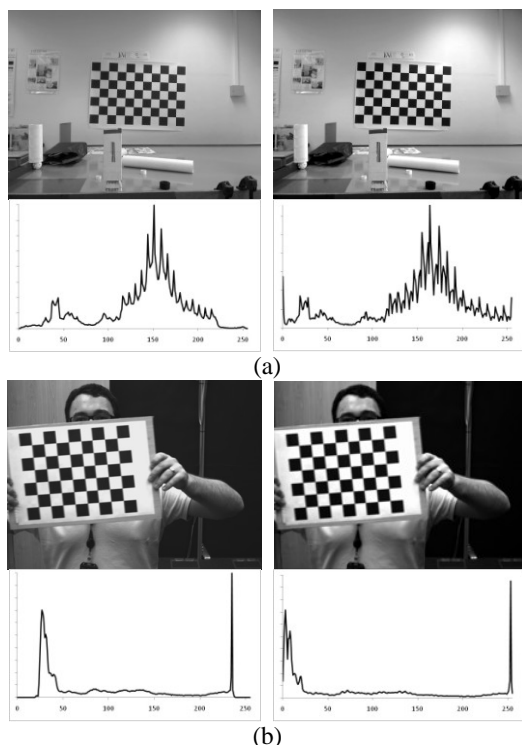


Figura 5. Mejora del contraste de las imágenes. En la columna izquierda se muestran las imágenes originales con sus histogramas y en la derecha el resultado de ampliar la escala.

Como se ha visto en los métodos anteriores, el tablero viene definido por los haces de líneas que son paralelas entre sí y perpendiculares entre ellas. La dificultad de basarse en los dos puntos de fuga más importantes en la imagen es que no se pueden utilizar imágenes en las que el patrón este casi perpendicular a la cámara ya que los puntos de fuga están de forma teórica en el infinito y de forma práctica se cortan en puntos muy alejados. Es por ello que en el método propuesto en este artículo se determinaran los haces de rectas mediante el patrón periódico que aparece en la transformada de Hough. Por otro lado al tenerse casos en los que la óptica tiene una distorsión geométrica importante, las rectas del patrón están curvadas en la imagen, por lo que la intersección de ellas no coincide con los puntos del patrón. Por esta causa se conjuga lo anterior con la detección de esquinas, por un lado para validar las rectas que pertenecen al patrón respecto a las del resto de objetos, por otro para determinar con precisión sub-píxel la verdadera posición de los puntos.

3. EXTRACCIÓN DE CARACTERÍSTICAS

3.1 Detección de esquinas

La calibración de una cámara es una operación en la que el usuario tiene el control de muchas variables del entorno, entre ella la iluminación. Por ello es de suponer que las imágenes estarán bien contrastadas. De todas formas, como esta etapa se va a diseñar pensando que los operarios no tengan una formación especializada y buscando el método más general posible, lo primero que se realiza es una mejora del contraste de las imágenes (Figura 5). Para conseguirlo, a partir del histograma de la imagen original se encuentran el nivel de gris más bajo, T_l , y más alto, T_h , que tengan un número mínimo de píxeles. A partir de estos valores, la transformación de los niveles de gris sigue la fórmula:

$$\begin{aligned} Lut(i) &= 0 & i < T_l \\ Lut(i) &= 255 \frac{i - T_l}{T_h - T_l} & T_l \leq i \leq T_h \\ Lut(i) &= 255 & T_h < i \end{aligned} \quad (3)$$

Siendo i el nivel de gris de la imagen original, la función Lut de la nueva y T_l y T_h los valores obtenidos para el negro y el blanco. Para encontrarlos se analiza el histograma y se obtiene el valor máximo, pasándose a examinar al histograma buscando el primer y último nivel de gris que tienen como mínimo un tanto por ciento prefijado del valor del máximo.

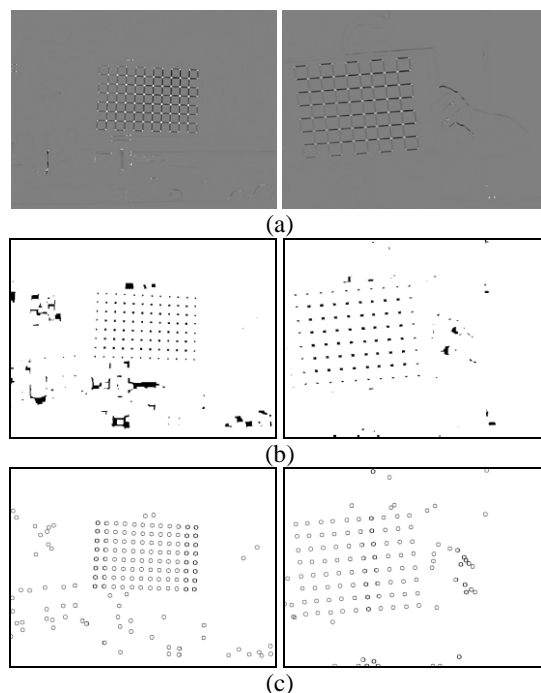


Figura 6. Detector de esquinas de Harris (a) imagen resultante (b) umbralización de los valores positivos (c) centro de gravedad de los objetos que no son ni muy pequeños ni muy grandes

El detector de esquinas de Harris es ampliamente utilizado para encontrar esta característica en las imágenes. Sin embargo puede utilizarse también para la detección de los puntos iniciales a la transformada de Hough ya que su respuesta es positiva para las

esquinas, casi nula para las zonas uniformes y negativa para los bordes.

Así, después de aplicar el detector de esquinas (Figura 6-a) se umbraliza la imagen para obtener las esquinas (Figura 6-b). Sobre todos los objetos de esta imagen se realiza un análisis dimensional rechazándose aquellos que tienen áreas por debajo, A_{min} , o por encima, A_{max} , de un umbral. Del resto se obtiene su centro de gravedad (Figura 6-c) que será una de las características que definen al tablero de ajedrez.

3.2 Transformada de Hough

La otra información que se va a utilizar es la intersección entre las principales rectas de la imagen. Para ello se realiza la transformada de Hough para rectas utilizando como información de entrada los puntos detectados como rectas de la imagen de Harris (Figura 7).



Figura 7. Detección de los puntos de bordes.

Los rangos de las variables serán:

$$\rho_1 = x \cos \theta_1 + y \sin \theta_1 \quad \left(\rho_1 \in [-\rho_{1max}, \rho_{1max}], \theta_1 \in \left[-\frac{\pi}{2}, \frac{\pi}{2} \right] \right) \quad (4)$$

$$\rho_{1max} = \frac{\sqrt{N^2 + M^2}}{2}$$

para el conjunto de coordenadas (x,y) detectado como puntos de un borde y donde N y M son las dimensiones horizontal y vertical de la imagen.

Como el objetivo de esta etapa es buscar los máximos de la transformada de Hough, en lugar de votar a un solo punto, se incrementa el contador de Hough alrededor del punto elegido siguiendo la siguiente matriz de votos:

1	2	1
2	4	2
1	2	1

Los resultados para las dos imágenes que se han mostrado hasta ahora pueden observarse en la Figura 8. Se aprecia como los máximos de la transformada se distribuyen espacialmente a lo largo de una línea desde la parte superior a la inferior de la imagen. Estos máximos serán la entrada a una segunda transformada de Hough que indicará las familias de rectas que hay en la imagen.

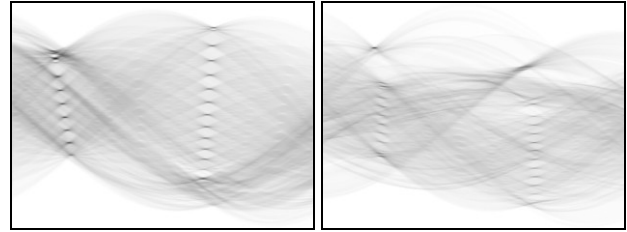


Figura 8. Transformada de Hough de la imagen de bordes.

Para conseguir esto, se deben de tener una serie de precauciones: la primera es decidir si se toma un número de máximos fijo o variable dependiendo del valor de la transformada. En ambos casos se trata de elegir todas las rectas que constituyan el patrón pero sin tomar muchas más rectas que no pertenezcan a él, ya que así se facilita el posterior análisis, como se muestra a continuación.

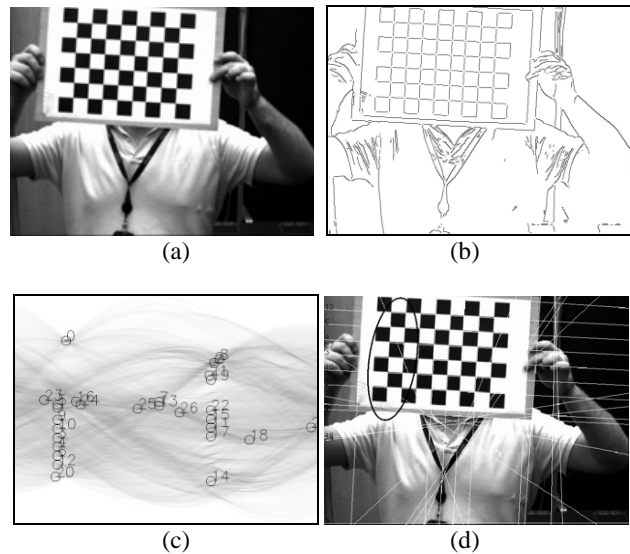


Figura 9. Resultados de tomar un número fijo de rectas. (a) Imagen original (b) imagen de bordes (c) transformada de Hough con los 36 máximos (d) rectas detectadas.

Así en la Figura 9 se tiene un caso en el que no se logran detectar todos los puntos. Aunque el tablero presenta 18 rectas (entre verticales y horizontales) y se han buscado el doble de ellas no se han encontrado todas, marcándose en la imagen la zona no detectada. El enfoque basado en buscar un número fijo de rectas dependería además del número de cuadros del patrón por lo que dependiendo de éste a veces se tendrían demasiadas y otras serían pocas. Es mejor por tanto un número de rectas que varíe con cada imagen.

Así, se pueden tomar tantos máximos a partir de un cierto umbral de puntos. Sin embargo, y como se ve en la Figura 10, este enfoque determina demasiadas rectas y puede llegar a complicar su posterior análisis. Los fallos son debidos a que se detectan máximos que realmente están en una zona llana, cuando las rectas que interesan vienen determinadas por ser máximos locales bien diferenciados de su entorno de vecindad.

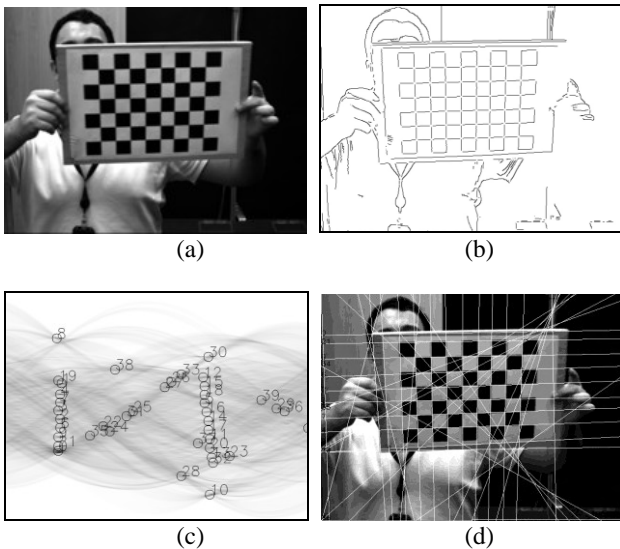


Figura 10. Resultados de tomar un número variable de rectas. (a) imagen original (b) imagen de bordes (c) transformada de Hough con los máximos (d) rectas detectadas.

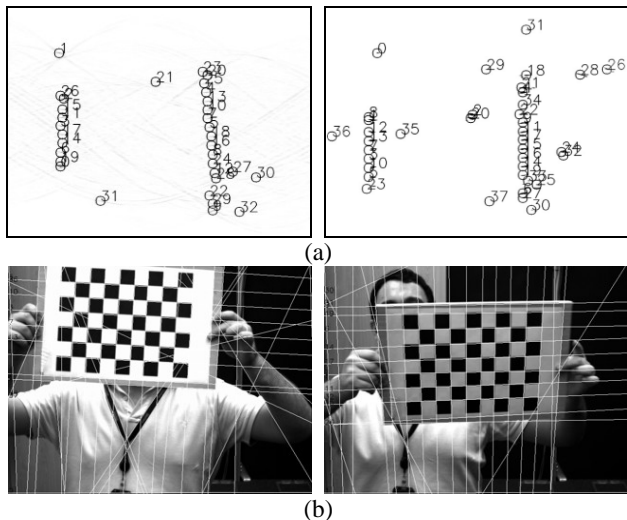


Figura 11. Rectas detectadas mediante la búsqueda de máximos locales. (a) máximos locales (b) rectas detectadas

Para encontrar estos puntos se ha utilizado la transformación morfológica “sombrero de copa” (*top-hat* en inglés), que consiste en realizar primero una transformación morfológica de “apertura” (*opening* en inglés) para luego restarla a la imagen original:

$$T = I - I \circ b \quad (5)$$

Donde I es la imagen original, T la resultante y b el elemento estructural con el que se realiza una erosión.

Como se observa en la Figura 11, los máximos resaltados corresponden a los máximos locales que se querían detectar. El número de rectas es variable y corresponde a aquellos máximos de la transformación morfológica que sean mayores a un valor prefijado. Para evitar detectar rectas muy próximas, después de obtener un valor se anula una zona alrededor del punto almacenado.

4. DETECCION DEL TABLERO

4.1 Segunda transformada de Hough

A partir de la Figura 11, puede observarse como muchos de los puntos detectados siguen la distribución de una recta. Se corresponden a líneas cuyo punto de corte sería el mismo y que forman un haz de rectas. Para detectarlas se vuelve a calcular la transformada de Hough con estos puntos (ρ_1, θ_1) como entradas.

$$\rho_2 = \theta_1 \cos \theta_2 + \rho_1 \sin \theta_2 \quad \left(\rho_2 \in [-\rho_{2\max}, \rho_{2\max}], \theta_2 \in \left[-\frac{\pi}{2}, \frac{\pi}{2} \right] \right) \quad (6)$$

$$\rho_{2\max} = \frac{\sqrt{4\rho_{1\max}^2 + \pi^2}}{2}$$

Tal como se muestra en la Figura 12 la segunda transformada de Hough presenta unos máximos que corresponden a las principales familias de rectas de la imagen.

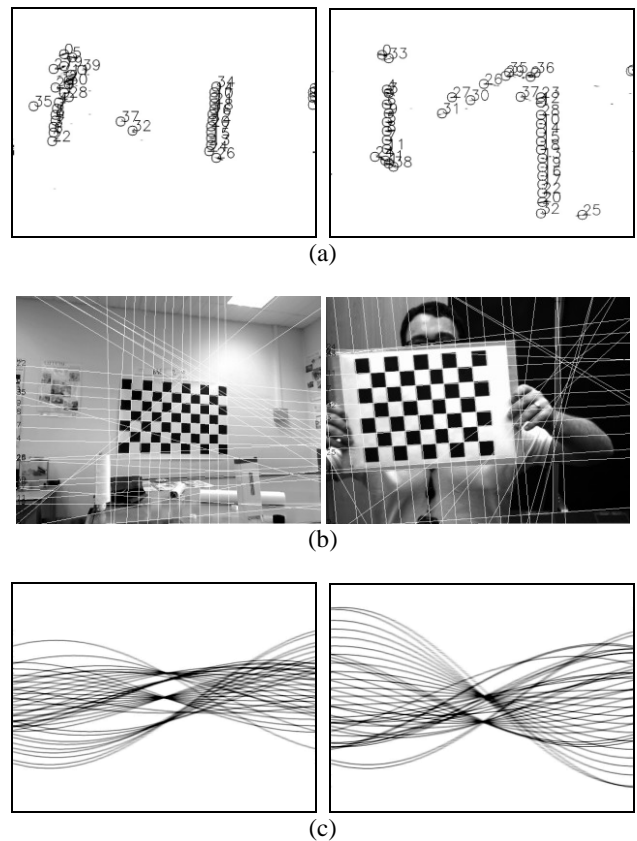


Figura 12. Detección de los haces de rectas. (a) Máximos de la transformada de Hough (b) imágenes con las rectas detectadas (c) segunda transformada de Hough.

En principio, con buscar los dos primeros máximos en esta segunda transformada de Hough sería suficiente, ya que el patrón está formado por dos haces de rectas. Sin embargo, y como se verá más adelante, los haces no se corresponderán siempre a rectas que estén paralelas en la imagen sino a rectas convergentes. Por ello se elegirá hasta un máximo de tres familias de rectas de las que luego se eliminará una. Por tanto se buscan los tres primeros máximos. Cada uno de ellos se considera como un posible haz de rectas si está por encima de un umbral. Para no elegir haces muy próximos se anula una zona

alrededor del punto obtenido. También hay que considerar que en la transformada de Hough los valores de ρ son positivos y negativos. Para evitarlo, se anula una zona que tenga el mismo ángulo pero distinto signo.

4.2 Determinación de los dos haces de rectas principales

Una vez encontrados al menos dos haces de rectas y como máximo tres, se pasa a determinar cuáles son los dos haces de rectas principales y qué rectas pertenecen a ellos. La solución más intuitiva para ello sería calcular la distancia de cada una de las rectas (ρ_1, θ_1) a la familia de rectas (ρ_2, θ_2)

$$dist = |\theta_1 \cos \theta_2 + \rho_1 \sin \theta_2 - \rho_2| \quad (7)$$

Los resultados pueden observarse en la Figura 13, donde se muestra un caso sencillo en el que el patrón está casi perpendicular a la cámara. Se ven las líneas detectadas y cómo es filtrada la recta que no es coherente con las demás. Sin embargo si solo se considera a la distancia dada por (7) se producirían numerosos fallos. Por ejemplo en la Figura 14 (d-h-l), se ha marcado el área de la imagen donde están las rectas no encontradas.

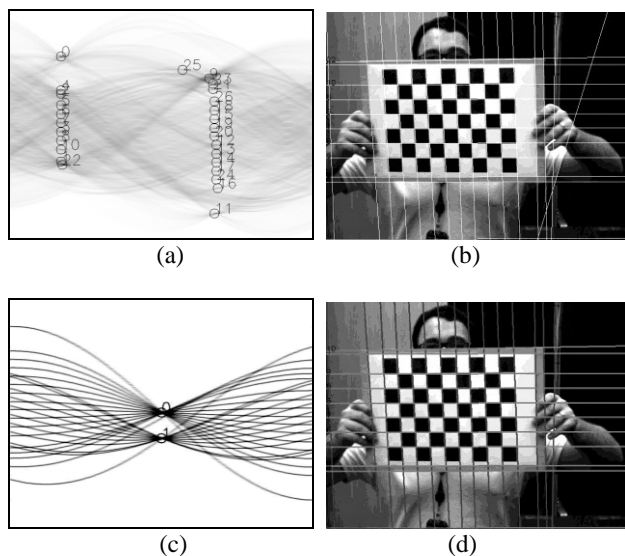


Figura 13. Detección inicial de familias de rectas

Estos errores son debidos a que no se ha tenido en cuenta el carácter periódico de la transformada de Hough (Figura 15-a). Así en este caso el ángulo de las rectas se ha tomado entre $\theta_1 \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ pero en otras implementaciones se toma entre $\theta_1 \in [0, \pi]$, siendo los resultados similares pero desplazados a la derecha.

Por lo tanto, se van a tener un conjunto de rectas (ρ_1, θ_1) que están alineadas, pero dado el intervalo en el que θ_1 está definido, no se encuentran representadas por los mismos valores de (ρ_2, θ_2) sino por $(-\rho_2, \theta_2)$.

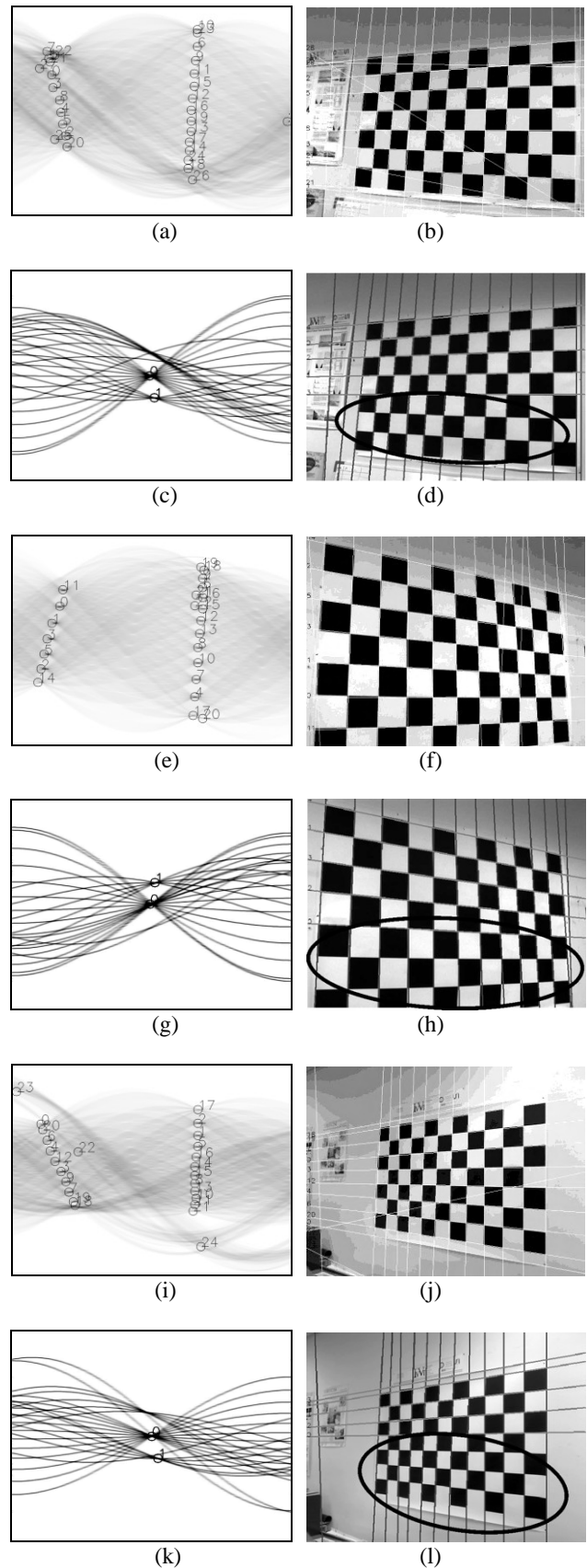


Figura 14. (a-e-i) Máximos de la transformada de Hough (b-f-j) rectas detectadas (c-g-k) familias de rectas (d-h-l) rectas cuya distancia según la ec. (7) es pequeña

Tal como se observa en la Figura 16, se han detectado más rectas. Sin embargo, no se han terminado de localizar todas las

que forman el patrón. Esto es debido a que éste no está formado solo por las que son paralelas sino también por las convergentes, por lo que también habrá que asociar éstas al haz detectado.

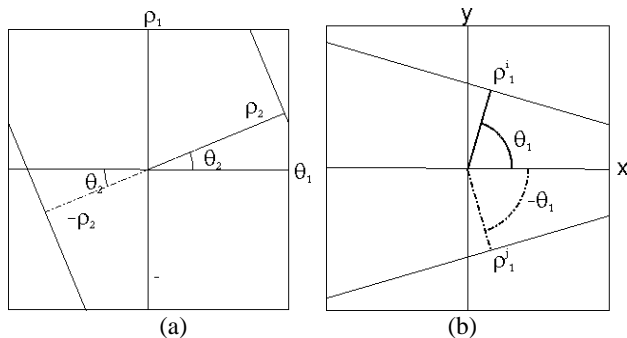


Figura 15. (a) Carácter periódico de la transformada de Hough
(b) Caso de rectas convergentes

Para analizar el caso de rectas convergentes se puede observar el supuesto de la Figura 15-b. Así, las rectas que sean paralelas a la recta i tendrán ángulos similares a θ_1 y vendrán asociadas a los valores (ρ_2, θ_2) y $(-\rho_2, \theta_2)$ de la segunda transformada de Hough. El caso de la recta j , que es convergente, tiene un valor parecido a θ_1 pero negativo por lo que las rectas paralelas a ella estarán alrededor de los valores $(\rho_2, -\theta_2)$ y $(-\rho_2, -\theta_2)$. Por tanto, para determinar el haz de rectas se calculan cuatro distancias, se toma la menor de ellas y si es menor que un umbral, prefijado de forma experimental, se asocia a ese haz (Figura 17).

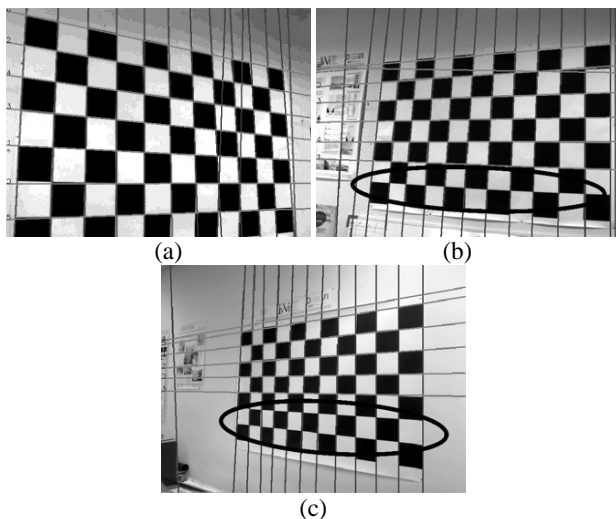


Figura 16. Detección de familias de rectas (a) detección correcta
(b-c) detecciones erróneas

4.3 Localización del patrón

De las tres familias de rectas anteriores se descarta la que tenga menos rectas asociadas. De las otras dos, una será la que corresponda a las rectas más horizontales y otra a las más verticales. Si se calcula el valor medio del ángulo:

$$\theta_i = \frac{\sum_{k=1}^{n_i} |\theta'_i|}{n_i} \quad i = (1, 2) \quad (8)$$

Se aprecia que aquella familia con el menor valor corresponde a las rectas que aparecen más verticales en la imagen.

Para las rectas que pertenecen al tablero de ajedrez debe cumplirse que los puntos de cruce entre las dos familias de rectas estén próximos a una de las esquinas que se encontraron con el detector de esquinas de Harris (Figura 6- c). Por lo tanto para cada punto de cruce entre rectas se analizan todas las esquinas detectadas, se elige la que esté más próxima y el punto de cruce se considera válido si la distancia está por debajo de un umbral, retirándose la esquina de la lista de ellas. Al acabar el proceso para todos los puntos de cruce, cada recta, tanto horizontal como vertical, tiene asociados una serie de intersecciones consideradas válidas.

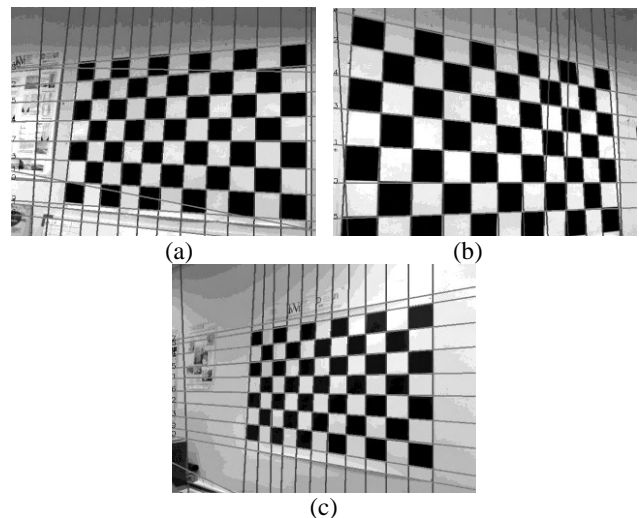


Figura 17. Detección de rectas

Una vez encontrados los puntos de cruce hay que descartar rectas, como la que se encuentra más a la izquierda en la figura 17-c. Éstas son rectas aisladas que están presentes en el entorno y que se caracterizan por tener pocos puntos de cruce. Como el tablero mínimo es de 3x3 puntos, todas aquellas rectas con menos de tres puntos de cruce son eliminadas. Quitar una recta afecta al número de puntos de cruce que siguen siendo válidos en las restantes, por lo que vuelve a calcularse el número de puntos de cruce válidos y el proceso de eliminación de rectas antes descrito hasta que este proceso iterativo no cambia al conjunto de rectas. Por lo tanto se realizan los siguientes pasos:

1. Obtener los puntos de cruce válidos entre rectas.
2. Eliminar las rectas con menos de tres puntos de cruce.
3. Si se ha eliminado alguna recta se vuelve al punto 1

El siguiente filtrado de rectas se basa en que las que pertenecen al patrón deben de tener los mismos puntos de cruce entre sí, ya que éstas corresponden al número de filas y columnas del tablero. Para ello:

1. Se calcula cual es la mediana del valor de puntos de cruce válidos para las dos familias de rectas. Ya que terminará determinando el número de columnas y filas.
2. En ambas familias de rectas se eliminan aquellas cuyos valores de puntos de cruce sean menores que las respectivas medianas
3. Si se ha eliminado alguna recta se vuelve al punto 1

Los dos conjuntos de rectas son entonces las líneas del tablero. En la Figura 18 se pueden ver varios resultados. Los círculos negros finos corresponden a las esquinas detectadas aplicando Harris. Los círculos con trazo negro grueso son aquellas intersecciones de rectas que tienen una esquina cercana y los de

trazo gris grueso son las intersecciones con las que no se corresponde a ninguna esquina.

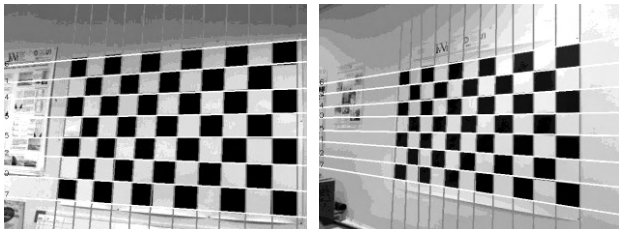
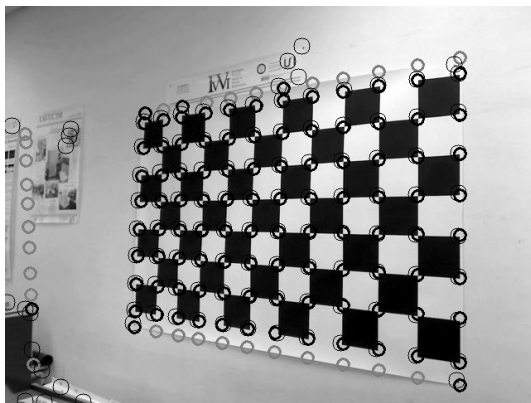
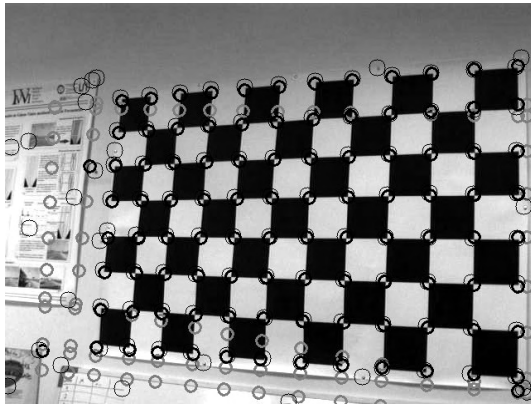


Figura 18. Emparejamiento de puntos de cruce y esquinas y rectas detectadas.

Una vez alcanzado que se tiene una serie de rectas hay que determinar cuáles son realmente las del tablero ya que puede haber alguna recta que sea paralela a los ejes del tablero pero que no pertenezca a éste. Para eliminarla se utiliza la información de que las rectas buscadas son la proyección en la imagen de un patrón conocido, en este caso un tablero de ajedrez del que se desconoce el número de filas y columnas pero sí se sabe que los cuadrados tienen unas dimensiones conocidas. Para ello se va a usar la razón doble de cuatro puntos alineados, A, B, C y D que se define:

$$RD(ABCD) = \frac{\overline{AC} \cdot \overline{AD}}{\overline{BC} \cdot \overline{BD}} \quad (9)$$

Sin perder la generalidad, se puede suponer que los cuatro puntos son consecutivos y que pertenecen a una línea horizontal, por lo que tienen la misma coordenada vertical:

$$\begin{aligned} A &= (x, y) & B &= (x + \Delta x, y) \\ C &= (x + 2\Delta x, y) & D &= (x + 3\Delta x, y) \end{aligned} \quad RD(ABCD) = \frac{2\Delta x \cdot 3\Delta x}{\Delta x \cdot 2\Delta x} = 3 \quad (10)$$

Siendo Δx la dimensión del cuadrado. Es importante señalar cómo el valor invariante no depende de él. Al estar el patrón constituido por cuadrados la razón doble de cuatro puntos verticales consecutivos tendría el mismo valor. El uso de la ecuación (9) estriba en que es un valor invariante a la proyección, luego cuatro puntos consecutivos del patrón siempre tendrán como razón doble el valor 3 aunque ya no aparezcan sobre una línea horizontal o vertical en la imagen.

Por lo tanto, se ordenan las rectas de izquierda a derecha y de arriba a abajo. Se calcula la razón doble para las cuatro primeras y si está dentro de un error del 5% del obtenido en (10) se consideran válidas. En caso contrario, se rechaza la primera recta y se añade la quinta volviéndose a calcular el valor. Así se irían tomando rectas hasta obtener las cuatro primeras que lo cumplan, que constituirían las rectas más a la izquierda del patrón. De forma análoga se tomarían las últimas cuatro rectas y se calcularía la razón doble y aceptando o rechazando y añadiendo otra recta se calcularía el extremo derecho del patrón. Siguiendo el mismo razonamiento para la otra familia de rectas se habría obtenido el contorno del tablero.

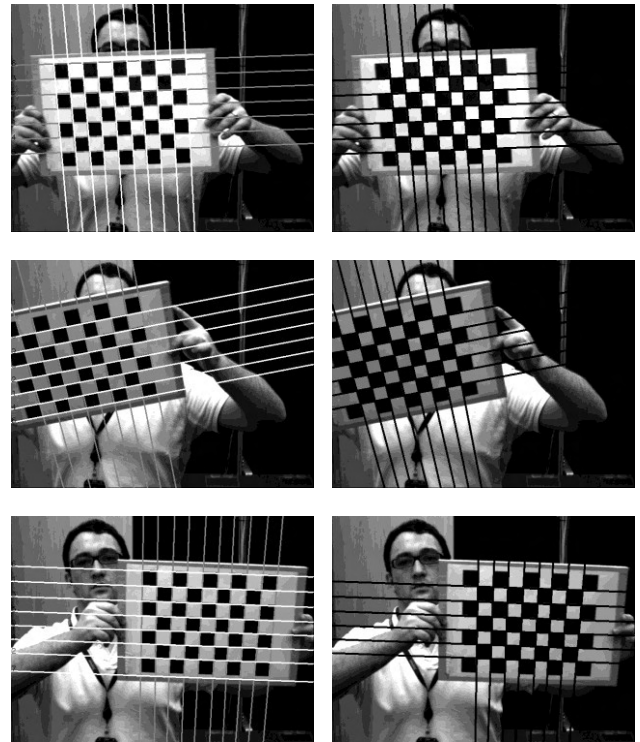


Figura 19. Filtrado final de rectas

El último paso es la eliminación de las rectas exteriores del tablero, de tal forma que el sistema detecte las de la zona interior del objeto. Esto es debido a dos razones:

1. Los contornos externos del tablero, al ser una línea en la que solo la mitad de los cuadrados aportan punto a la transformada de Hough, no se detectan siempre. Por ello si se quiere hacer una detección robusta ante distintas posiciones y condiciones no se puede añadir un elemento cuya detección sea incierta. Aunque puede solucionarse, la siguiente razón desaconseja su uso
2. Cómo la detección del tablero sirve para la detección de esquinas, el borde de éste es donde las esquinas están peor definidas ya que no se da una alternancia blanco-negro en

dos direcciones como en el resto. Por ello no son buenos puntos para la calibración.

Para comprobar si se tiene que eliminar o no, se observa el valor de la transformada de Hough y se compara con el ancho del patrón obtenido. Si no supera un umbral es que se trata de una línea discontinua. Ese proceso se realiza para los cuatro bordes del patrón. Algunos resultados de la detección final se ven en la Figura 20.



Figura 20. Puntos de entrada al algoritmo de calibración.

Por último, y debido a la distorsión geométrica los cruces entre las rectas no van a ser los valores de las esquinas por lo que se toman estos como valores iniciales, buscándose las esquinas en un entorno de ellos, constituyendo los punto de entrada al algoritmo de calibración (Figura 20).

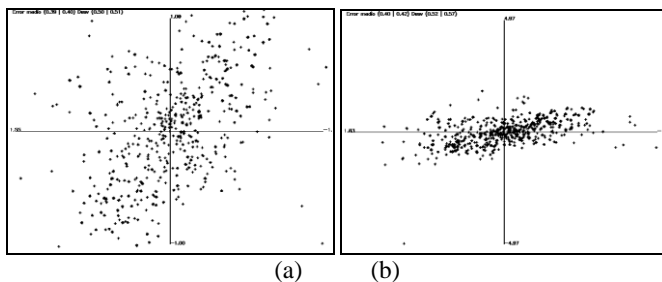


Figura 21. Errores entre la proyección teórica y real de los puntos del patrón en la imagen para (a) la calibración inicial (b) refinando los resultados.

5. CALIBRACION DE LAS CAMARAS

5.1 Calibración de parámetros intrínsecos

El sistema se calibra utilizando la implementación del algoritmo de Zhang desarrollada en las *OpenCV* y que da como salida los parámetros intrínsecos de la cámara, teniendo en cuenta toda la secuencia de imágenes. Una vez obtenidos estos valores se pueden calcular los parámetros extrínsecos que ayudarán a depurar los valores intrínsecos.

Para ello se proyectan sobre la imagen los puntos del patrón. Siguiendo la ecuación (1) se conocen las coordenadas del mundo \mathbf{X} por lo que se puede obtener los puntos \mathbf{U} . Mediante la ecuación (2) se puede deshacer el error de la distorsión por lo que ya se tiene el valor real de la proyección del punto y el valor teórico. Esto permite calcular el error cometido (Figura 21-a) del que se obtiene su valor medio, μ_e , y desviación típica, σ_e . Se desechan todos aquellos puntos cuyo error caiga fuera del intervalo $(\mu_e - 3\sigma_e, \mu_e + 3\sigma_e)$. Se vuelve a calibrar el sistema con

los puntos que están dentro de este intervalo. Si se calcula el error cometido ahora (Figura 21-b) se observa que éste es menor.

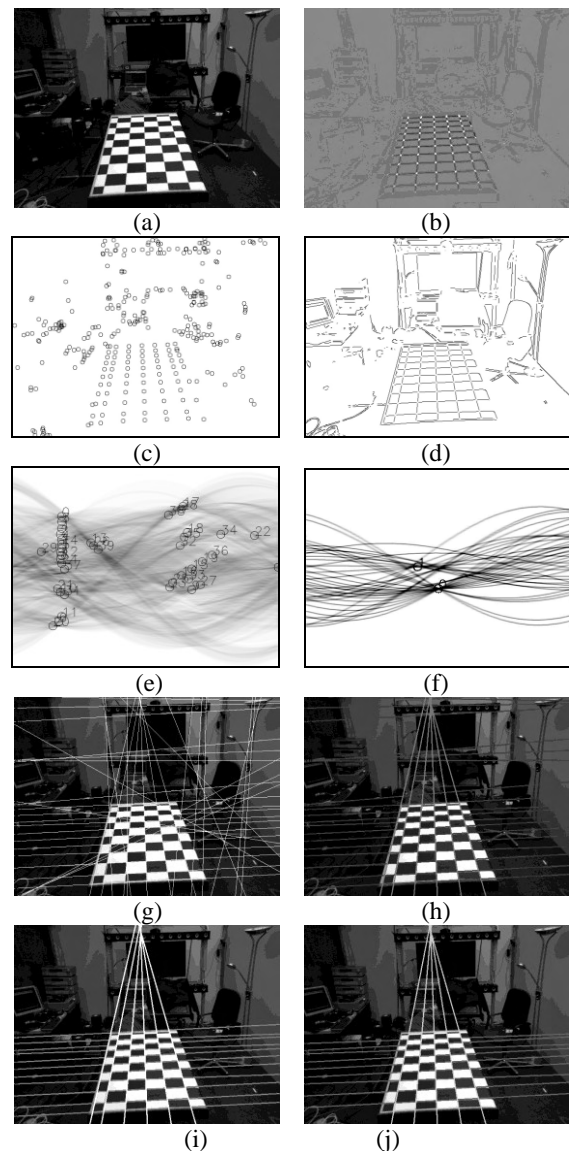


Figura 22. (a) imagen original (b) operador de Harris (c) esquinas detectadas (d) bordes de los objetos (e) transformada de Hough (f) transformada de Hough de las principales rectas (g) principales rectas (h) familias de rectas (i) rectas coherentes con las esquinas detectadas (j) rectas finales (k) tablero

5.2 Calibración de parámetros extrínsecos

El sistema descrito no sirve solamente para la calibración de los parámetros intrínsecos, sino que también permite calcular la posición y orientación de la cámara respecto al sistema global de referencia. Tal como se observa en la Figura 22 se coloca un tablero de dimensiones adecuadas en una posición conocida. El sistema lo detecta y sitúa la cámara dentro del sistema de referencia global. Este caso es más complejo que el de la calibración de parámetros intrínsecos ya que en la imagen existen muchos más objetos al encontrarse el patrón más alejado y no abarcar gran parte de ella. Se ve así que el número de esquinas es mucho mayor (Figura 22-c)

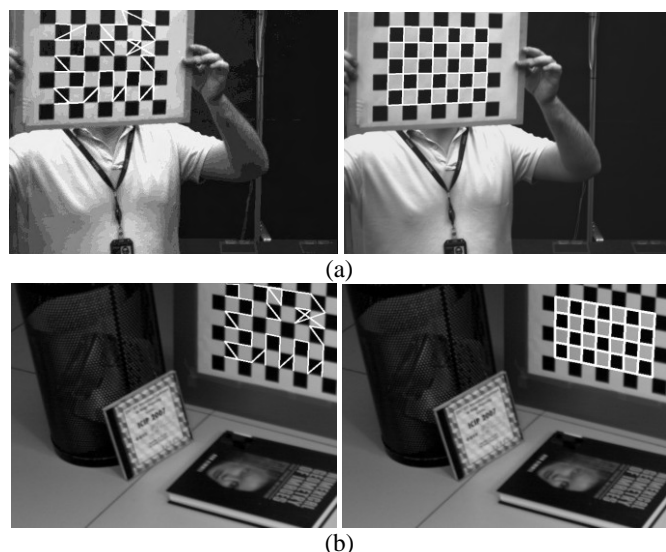


Figura 23. Comparación entre los resultados de la detección del tablero según las (a) OpenCv y (b) el método propuesto

6. RESULTADOS

Para mostrar la robustez de los resultados del sistema propuesto se va a comparar con métodos ya establecidos, tanto para el caso de la detección del tablero como para los valores de calibración.

Las OpenCV realizan una umbralización seguida de una erosión para romper la conectividad de los cuadros pasando a buscar su número y posición en la imagen. Como el número de filas y columnas es una de las informaciones que hay que pasar, esta erosión se repite el número de veces necesaria para romper la conectividad y encontrar ese número. Después, los cuadrados se vuelven a unir para formar el tablero. En la Figura 23 se ven dos casos en que la detección por las OpenCv no ha funcionado. En la Figura 23-a el patrón es perfectamente visible pero una de las esquinas elegidas no pertenece al patrón sino al borde de la

imagen. En la Figura 23-b se ve el caso de la oclusión de una esquina que hace imposible que las OpenCv encuentren el patrón. Como en el método propuesto el número de filas y columnas no está determinado no hay problema para que encuentre un tablero con una fila menos que el real. Hay que hacer notar que esto no produce errores en el proceso de calibración ya que lo que le importa a éste es la posición relativa entre los puntos. Como conclusión, en las OpenCv hay que pasar expresamente para cada calibración el número de puntos y el patrón tiene que verse totalmente en la imagen, a diferencia del método propuesto.

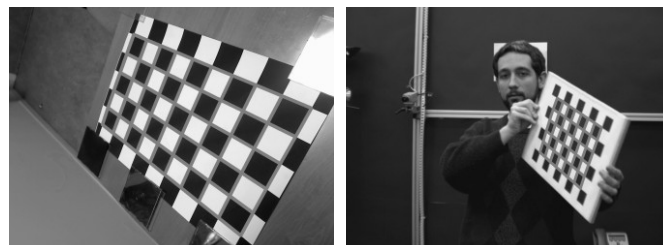


Figura 24. Giros

En la figura 24 se presentan varios casos en los que el patrón de calibración está bastante girado para mostrar que el método es robusto ante la orientación del propio patrón o de la cámara

Los resultados de la calibración con este método se han comparado para los obtenidos con el Toolbox de Matlab de Bouguet para tres secuencias y se muestran en la tabla 1. Como se observa los resultados son similares y el error cometido está dentro de los mismos márgenes, con la diferencia de que con el método propuesto no hay que marcar cada una de las esquinas del tablero de ajedrez por lo que la tarea de calibración se hace mucho menos tediosa y propensa a fallos.

Respecto a los parámetros extrínsecos, en la Figura 25 se muestra una imagen en la que después de eliminar la distorsión de ella se han superpuesto las esquinas de la habitación que se han proyectado sobre la imagen a partir de los parámetros intrínsecos y extrínsecos. Se observa como los resultados son correctos.

Los tiempos de ejecución en este tipo de tareas no son críticos al tratarse de un proceso de *off-line*, aunque si son muy altos si que pueden ser importantes si hay muchas cámaras que calibrar. Se ha probado en dos equipo distintos y sus resultados son para imágenes de 1024x768 píxeles: 1.9 segundos en un Intel Core2 Duo Mobile Processor a 1.66GHz con 2GB de RAM y 1.1 segundos en un Intel Core2 Duo a 3.00GHz con 3,25GB de RAM.

Tabla 1. Comparación de los valores de calibración entre Toolbox de Matlab de Bouguet y el método propuesto (MP)

		fx (px)	fy (px)	cx (px)	cy (px)	Distorsión radial		Distorsión tangencial		Error x (px)	Error y (px)
						k1 10 ⁻³	k2 10 ⁻³	p1 10 ⁻³	p2 10 ⁻³		
Secuencia 1	Matlab	821.1	812.8	327.4	234.6	-248.5	214.9	2.5	-5.1	0.50	0.36
	MP	820.6	812.4	327.4	234.7	-247.0	210.9	2.5	-5.1	0.39	0.40
Secuencia 2	Matlab	514.3	515.9	322.1	243.0	23.9	-157.5	3.1	1.5	0.30	0.17
	MP	517.3	518.8	320.6	239.8	3.1	-54.4	1.4	0.7	0.24	0.23
Secuencia 3	Matlab	2371.6	2421.8	326.3	389.8	-35.7	684.7	-3.96	-57.5	1.17	0.31
	MP	2440.0	2493.0	328.2	387.9	28.0	334.9	-4.2	-59.5	0.93	0.39



Figura 25. Dibujo del contorno de la habitación sobre la imagen capturada por la cámara

7. CONCLUSIONES

En el presente artículo se ha presentado un método capaz de detectar un patrón artificial en una imagen. Con el solo conocimiento previo de que se trata de un tablero de ajedrez, el algoritmo es capaz de determinar su posición en la imagen y averiguar el número de filas y columnas. A través de varios ejemplos se ha demostrado que el algoritmo es útil para las tareas de calibración que y su utilización hará menos tediosa esta tarea así como la pondrá al alcance de personal no especializado.

AGRADECIMIENTOS

Este trabajo ha sido realizado parcialmente gracias al apoyo del ministerio español de Ciencia e Innovación a través del CDTI y el proyecto CENIT-VISION 2007-1007.

REFERENCIAS

- Ahn S. J. Rauh W. Kim S. I. (2001). Circular coded target for automation of optical 3D-measurement and camera calibration. In: *International Journal of Pattern Recognition and Artificial Intelligence*, **15** (6), pp. 905-919.
- Bouguet J.Y. Camera calibration toolbox for Matlab. www.vision.caltech.edu/bouguetj/calib_doc
- Bradski G. and Kaehler A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly.
- Douskos V. Kalisperakis I. Karras G. (2007). Automatic calibration of digital cameras using planar chess-board patterns. In: *Optical 3-D Measurement Techniques VIII*, **1**, pp. 132-140.
- Douskos V. Kalisperakis I. Karras G. Petsa E. (2008). Fully automatic camera calibration using regular planar patterns.

- In: *International Archives of Photogrammetric, Remote Sensing and the Spatial Information Sciences*, **37** (5), pp. 21-26.
- Fiala M. and Shu Ch. (2008). Self-identifying patterns for plane-based camera calibration. In: *Machine Vision and Applications*, **19**, pp. 209-216.
- Forbes K. Voight A. Bodika N. (2002). An inexpensive automatic and accurate camera calibration method. In: *Proceedings of the Thirteenth Annual South African Workshop on Pattern Recognition*.
- Grammatikopoulos L. Karras G. Petsa E. Kalisperakis I. (2006). A unified approach for automatic camera calibration from vanishing points. In: *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, XXXVI, Part 5.
- Harris C. and Stephens M.J. (1988). A combined corner and edge detector. In: *Alvey Vision Conference*, pp. 147-152.
- Matas J. Soh L.M. and Kittler J. (1997). Object recognition using a tag. In: *Int Conf on Image Processing*, pp. 877-880.
- Ronda J. I. Valdés A. Gallego G. (2008). Line geometry and camera autocalibration. In: *Journal of Mathematical Imaging and Vision*, **32** (2), pp. 193-214.
- Shu C. Brunton A. and Fiala M. (2003). Automatic grid finding in calibration patterns using Delaunay triangulation. In: *Technical Report NRC-46497/ERB-1104*. National Research Council, Institute for Information Technology.
- Trucco E. Verri A. (1998). *Introductory techniques for 3-D Computer Vision*. Prentice Hall
- Tsai R.Y. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. In: *Journal of Robotics and Automation*, **3**, pp. 323-344.
- VISIÓN: Comunicaciones de Vídeo de Nueva Generación www.cenit-vision.org/index.php
- Wang Z. Wu W. Xu X. Xue D. (2007). Recognition and location of the internal corners of planar checkerboard calibration pattern image. In: *Applied mathematics and Computation*, **185** (2), pp: 894-906.
- Yu C. and Peng Q. (2006). Robust recognition of checkerboard pattern for camera calibration. In: *Optical Engineering*, **45** (9), 093201.
- Zhang Z. (2000). A flexible new technique for camera calibration. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22** (11), pp. 1330-1334.
- Zhang Z. (2004) Camera Calibration With One-Dimensional Objects. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26** (7), pp. 892-899.
- Zhaoxue C. and Pengfei S. (2004). Efficient method for camera calibration in traffic scenes. In: *Electronics Letters*, **40**, pp: 368-369.