

# Pasos críticos en la estimación de pose en cámara: una evaluación usando la biblioteca LTI-LIB2

Critical steps in camera pose estimation: an evaluation using LTI-LIB2 library

*Laura Cabrera-Quirós<sup>1</sup>  
Rafael Campos-Gómez<sup>2</sup>  
Jorge Castro-Godínez<sup>3</sup>*

*Fecha de recepción: 29 de julio del 2013  
Fecha de aprobación: 13 de setiembre del 2013*

Cabrera-Quirós, L; Campos-Gómez, R; Castro-Godínez, J. Pasos críticos en la estimación de pose en cámara: una evaluación usando la biblioteca LTI-LIB2. *Tecnología en Marcha*. Número Especial. Pág 60-69.

- 1 Ingeniera en Electrónica. Profesora en el Instituto Tecnológico de Costa Rica (ITCR). Estudiante del Programa de Maestría en Electrónica del ITCR. Correo electrónico: lcabrera@ietec.org
- 2 Ingeniero en Electrónica. Socio fundador de Altus Consulting S.A. Estudiante del Programa de Maestría en Electrónica del Instituto Tecnológico de Costa Rica. Correo electrónico: rcampos@altus.cr
- 3 Ingeniero en Electrónica. Profesor en el Instituto Tecnológico de Costa Rica (ITCR). Estudiante del Programa de Maestría en Electrónica del ITCR. Correo electrónico: jcastro@ietec.org

## Laura Cabrera Quirós

Cursó sus estudios de Licenciatura en Ingeniería Electrónica en el Instituto Tecnológico de Costa Rica y posteriormente su Maestría en Electrónica con énfasis en sistemas empotrados, siendo parte de la primera generación de graduados de dicha maestría. A partir de junio de 2012 se incorpora como profesora de la Escuela de Ingeniería de Electrónica

y como apoyo para el área de Ingeniería en Computadores. En 2014 es becada por el Instituto Tecnológico de Costa Rica para cursar sus estudios doctorales en la TU Delft, Holanda. Sus áreas de especialidad se enfocan en el procesamiento de imágenes y reconocimiento de patrones, para aplicaciones de visión por computadora.

## Palabras clave

ASM; calibración de cámara; esquinas del tablero de ajedrez; estimación de pose; LTI-LIB2.

## Resumen

En este artículo se presenta una evaluación de métodos de estimación de pose en cámara, usando un patrón de tablero de ajedrez. Los pasos evaluados en el proceso de estimación son la detección de puntos de interés y la estimación de parámetros de cámara, debido a su criticidad en el proceso como un todo. Se comparan el método ChESS y un método heurístico para la detección de patrones de tableros de ajedrez. Ambos métodos son objetivamente contrastados usando los criterios de Verdadero Positivo y Falso Negativo. Mientras tanto, el método de Zhang para estimación de pose en cámaras basado en distribuciones de puntos en superficies planas se usa como primera aproximación y luego es refinado por medio de una regresión no lineal usando el algoritmo de Levenberg-Marquardt. Este algoritmo de estimación de pose de cámara es evaluado por medio de una comparación con una herramienta estable, como lo es el paquete "Camera Calibration Toolbox" para Matlab®.

## Key words

ASM; camera calibration; chessboard corners; pose estimation; LTI-LIB2.

## Abstract

An evaluation of camera pose estimation methods using a chessboard pattern is presented. Steps evaluated in the estimation process are landmark point detection and camera parameter estimation, due to their critical role in the entire process. The ChESS method and a custom heuristic method are compared for chessboard pattern detection. Both methods are objectively contrasted using True Positive and False Negative criteria. Meanwhile, Zhang's method for pose estimation based on planar surface point distribution is used as a first approach, and then refined with a nonlinear regression through the Levenberg-Marquardt algorithm. This pose estimation algorithm is evaluated through a comparison with a stable tool, such as the Camera Calibration Toolbox for Matlab®.

## Introducción

Un gran número de aplicaciones de visión por computadora basan sus resultados en un correcto proceso previo de calibración de cámara. Pequeños errores introducidos por problemas de calibración pueden afectar significativamente los resultados finales, por lo que el proceso de calibración debe ser lo suficientemente exacto a fin de minimizar los potenciales errores inherentes a dichas aplicaciones.

Una de estas aplicaciones es la realidad aumentada, la cual consiste en la superposición de objetos generados por computadora sobre imágenes captadas en tiempo real. En general, todos los métodos de calibración estiman los parámetros intrínsecos y extrínsecos de la cámara, así como otras variables (como la distorsión de la lente) de manera opcional. Estos parámetros definen la forma en que objetos en tercera dimensión (3D) son proyectados en el plano de una imagen, por lo que su cálculo de manera correcta es vital para lograr que los objetos virtuales se acoplen de forma realista a la imagen real, también llamada escena.

El proceso de calibración se puede dividir generalmente en dos pasos críticos: 1) la detección de puntos de interés en imágenes de muestra y 2) la estimación de parámetros a partir de los puntos detectados. Este artículo busca comparar dos métodos para evaluar objetivamente estos dos pasos, como una primera aproximación para aplicaciones de realidad aumentada exactas aplicadas a sistemas embebidos. Dicha evaluación se realizará utilizando las herramientas de software para procesamiento de imágenes de la biblioteca LTI-LIB2 (Alvarado, 2012), codesarrollada por el Dr. Ing. Pablo Alvarado Moya, profesor de la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica (ITCR).

Primeramente, se comparan dos algoritmos de detección de puntos de interés para un tablero de ajedrez (uno de los patrones estándar para calibración de cámaras): el algoritmo ChESS (Bennet, 2013) y un algoritmo heurístico basado en la transformada de Hough. El resultado de ambos algoritmos es luego refinado utilizando el algoritmo de Modelo Activo de Formas (ASM, por sus siglas en inglés) desarrollado por Cootes (Cootes, 1995). Seguidamente, se evalúa la estimación de pose propuesta por Zhang para puntos en una superficie

plana (Zhang, 2000) aplicando optimización para el refinamiento de parámetros por medio de una regresión no lineal, recomendada por el autor.

El objetivo de esta evaluación es determinar el mejor algoritmo para la detección de puntos de interés para calibración de cámara por medio de un tablero de ajedrez y, además, evaluar la efectividad del algoritmo desarrollado por Zhang (2000) para la estimación de parámetros de calibración de una cámara a partir de imágenes planas, todos implementados utilizando las herramientas para procesamiento de imágenes de la biblioteca LTI-LIB2 y manteniendo la perspectiva de futuras aplicaciones para sistemas embebidos. En los antecedentes se resumen trabajos previos de importancia en el área de interés. Luego se detalla la propuesta de comparación de métodos de detección de puntos y la evaluación del algoritmo de estimación de pose, así como el protocolo experimental aplicado. Seguidamente se resume los resultados encontrados. Finalmente, se presentan las conclusiones y el trabajo futuro para esta investigación.

## Antecedentes

La detección de un patrón es el punto de inicio fundamental en un proceso de calibración de cámara. La detección de esquinas es una técnica ampliamente utilizada, cubierta desde diferentes enfoques, para obtener puntos conocidos en el patrón de calibración, usualmente un tablero de ajedrez. En su artículo, de la Escalera y Armingol (2010) primero aplican un detector de esquinas de Harris, seguido de un umbral y la determinación de centros de gravedad para obtener las esquinas. Luego de esto, se aplica la transformada de Hough a la imagen y se realiza una detección de líneas por medio de una representación en frecuencia de máximos locales. Una segunda transformada de Hough se aplica junto con una transformación morfológica para detectar las líneas que corresponden al tablero de ajedrez. En la localización del patrón del tablero de ajedrez, se determinan las intersecciones entre las líneas que coinciden con las esquinas encontradas mediante Harris, de manera que en aquellos casos que se encuentre una relación geométrica de 4 puntos alineados se garanticen puntos del tablero, y con esto descartar aquellos puntos que no corresponden al patrón buscado.

Kassir y Paynot (2010) proponen en su método aplicar una mejora de contraste adaptable y luego un detector de esquinas de Harris para tres tamaños de ventana diferentes. El conjunto más grande de esquinas se escoge para ajustarse a una cuadrícula y luego interpolar las esquinas no detectadas. Este método comprobó buenos resultados aún con cámaras con lentes distorsionados (del tipo “fisheye”, por ejemplo).

Tam y compañía (2008) presentan un método usando detección de esquinas de Harris y SUSAN, comparando cada píxel, unidad básica del procesamiento de imágenes, con sus vecinos y una transformada de Radon para estimar las líneas e intersecciones de las líneas del tablero de ajedrez.

Por otra parte, se han propuesto varios métodos para calcular los parámetros intrínsecos y extrínsecos de una cámara. El caso general se presenta cuando los puntos en el mundo real tienen posiciones arbitrarias en el espacio en tercera dimensión (3D). Un caso menos general, pero más práctico, se da cuando los puntos reales se restringen a una superficie plana, como es el caso para un tablero de ajedrez. Cuando éste es el caso, el método más extendido de calibración basado en superficie plana lo propuso Zhang (2000). Dicho método necesita solamente tres vistas del patrón, en diferentes orientaciones, para el cálculo y no requiere conocimiento sobre el movimiento de la cámara. Incluso, la solución es dada de manera cerrada y puede calcularse utilizando métodos lineales simples como la descomposición de valores singulares (SVD, por sus siglas en inglés). Un refinamiento no lineal es sugerido por el autor para compensar el ruido en los puntos de la imagen y para incorporar el efecto de la distorsión radial del lente de la cámara.

La biblioteca LTI-LIB2 para el lenguaje C++, orientada a objetos en su totalidad, es escogida para la evaluación pues ha probado ser para los autores una herramienta versátil y potente, de acceso libre y con mantenimiento regular, que cumple con los

requerimientos mínimos necesitados para la implementación de los algoritmos que se van a evaluar.

## Propuesta de evaluación

Como ya se mencionó, el proceso de calibración de una cámara consta de dos etapas secuenciales fundamentales, como se muestra en la figura 1, cuyo funcionamiento afecta de manera significativa el resultado final: la detección de patrón para calibración (esquinas de tablero de ajedrez, en este caso) y la estimación de los parámetros extrínsecos e intrínsecos de la cámara. Cada etapa se evaluará por separado.

### Detección de esquinas en el tablero

Para esta etapa se busca comparar el método ChESS (Bennet, 2013) y un método heurístico para la obtención de las esquinas en un tablero de ajedrez, ambos basados en técnicas de procesamiento de imágenes. Cabe destacar que se busca un método capaz de discernir entre esquinas pertenecientes a un tablero de ajedrez de aquellas esquinas pertenecientes al fondo de la imagen, de manera independiente de la rotación o escala del tablero, la iluminación de la escena o la cámara utilizada. Ambos algoritmos buscan posibles esquinas de un tablero, basadas en la imagen pero no asignan probabilidad de pertenencia a estas posibles esquinas. Por esta razón, ambos algoritmos se complementan con el uso de un modelo activo de forma o ASM (Cootes, 1995) previamente entrenado con el modelo de un tablero de ajedrez regular. El resultado de esta etapa será un vector de 81 puntos de interés que representa la posición de las 81 esquinas en un tablero de ajedrez.

En cuanto a los métodos, ChESS (Bennet, 2013) realiza una evaluación de pertenencia a una posible esquina de tablero para cada uno de los píxeles en la imagen. Por otro lado, el método heurístico propuesto se basa en la transformada de Hough,



Figura 1. Etapas del proceso de calibración de la cámara.

ampliamente utilizada para la detección de líneas en imágenes (González, 2008), para la extracción de segmentos de línea en la imagen y consecuentemente evaluar los puntos en que dichos segmentos se intersecan entre sí. Con base en esto puede catalogarse al algoritmo heurístico como una optimización del algoritmo ChESS, en términos de rendimiento, lo cual se busca comprobar o rechazar. La evaluación en cada punto, sin embargo, difiere entre ambos algoritmos.

El algoritmo ChESS encuentra los vértices estableciendo ocho puntos alrededor del punto de interés, donde cada punto encuentre otro en una posición opuesta (a 180°), con las mismas características en sus píxeles. Por su parte, el algoritmo heurístico, para cada punto de intersección encontrado con la transformada de Hough, evalúa su pertenencia o no al patrón del tablero de la siguiente manera:

1. Se calcula el ángulo  $\alpha$  entre las dos líneas intersecadas en ese punto.
2. Se considera un círculo  $C$  de radio  $R$  centrado en el punto de intersección. Inicialmente en el ángulo  $\theta=0^\circ$ , realizando incrementos de  $5^\circ$ , se calcula la distancia euclidiana  $d_1$  en el espacio RGB entre el píxel localizado a una distancia  $R$  de la intersección con el ángulo  $\theta$  y el punto localizado diametralmente opuesto.
3. Para cada incremento de  $5^\circ$ , se calcula la distancia en el espacio RGB entre el píxel localizado a una distancia  $R$  del punto de intersección en la dirección  $\theta$  y el píxel en el círculo  $C$  en la dirección  $\theta+\alpha$ . Esta será la distancia  $d_2$ .
4. Si  $d_2$  es suficientemente alto y  $d_1$  suficientemente pequeño, el punto corresponde a una esquina del tablero. Un umbral para las distancias se implementa para lograr mejores resultados. Si este criterio se cumple para al menos la mitad de los ángulos probados, el punto se clasifica como parte del patrón, de lo contrario se rechaza.

### Estimación de pose de cámara

Los parámetros intrínsecos (centro de imagen, razón de aspecto de píxeles y sesgo) y extrínsecos (rotación y traslación) permiten conocer la posición de un punto de una imagen plana en el espacio real por medio de la Ecuación 1 (Szeliski, 2010), donde

$\mathbf{X}$  representa los puntos en el espacio real y  $\mathbf{x}$  los puntos en la imagen.

$$\mathbf{x} = \mathbf{P}\mathbf{X} \quad (1)$$

La matriz  $\mathbf{P}$ , llamada matriz de proyección, está formada por los parámetros intrínsecos y extrínsecos, según la Ecuación 2, donde  $\mathbf{K}$  representa la matriz de calibración intrínseca,  $\mathbf{R}$  la matriz de rotación y  $\mathbf{t}$  el vector de traslación, ambos parámetros extrínsecos.

$$\mathbf{P} = \mathbf{K} [\mathbf{R} | \mathbf{t}] \quad (2)$$

Para el vector de 81 puntos generado en la etapa previa se extraen cuatro puntos de referencia del tablero, específicamente los puntos mostrados en la Figura 2 pues, al encontrarse en el centro del patrón del tablero, poseen mayor confiabilidad de detección.

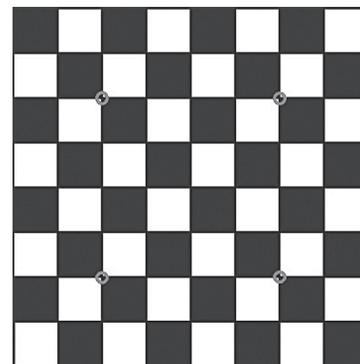


Figura 2. Puntos de referencia para estimación de pose.

Estos cuatro puntos se utilizan para estimar los parámetros intrínsecos y extrínsecos de la cámara por medio del método de calibración a partir de superficie plana propuesto por Zhang (2000). La matriz de parámetros intrínsecos  $\mathbf{K}$  es calculada usando tres imágenes con diferentes posiciones del tablero y sin alterar la posición de la cámara, este parámetro se mantiene mientras no se modifique el enfoque de la cámara, pues los parámetros intrínsecos son constantes. Para la aplicación en video, ésta se obtiene con al menos un segundo de retardo entre tres capturas. Una vez que la matriz de calibración intrínseca es calculada, la matriz de rotación  $\mathbf{R}$  y el vector de traslación  $\mathbf{t}$  se estiman para cada imagen o para cada cuadro, en el caso de aplicación en video.

En su investigación, Zhang (2000) recomienda que tanto el vector de traslación como las matrices de rotación y de calibración intrínseca se refinen utilizando regresión no lineal. Para este trabajo, se hace uso del algoritmo Levenberg-Marquardt (Szeliski, 2010). No es necesario realizar el refinamiento para  $K$  a menos que se altere el enfoque de la cámara. El vector de traslación y la matriz de rotación deben refinarse en cada imagen. Para efectos de esta evaluación, se parte de la utilización de lentes con distorsión despreciable y ésta no se incorpora a los parámetros de calibración.

Para el proceso de evaluación se comparan los resultados del método de Zhang (2000) contra resultados generados por la herramienta "Camera Calibration Toolbox" para Matlab®, comprobados previamente como correctos y estables.

## Resultados

### Detección de esquinas en tablero

Se utilizó un conjunto de 20 imágenes para comparar los algoritmos de detección de esquinas propuestos. La detección fue realizada para las esquinas interiores del tablero, las esquinas exteriores no se tomaron en consideración, resultando en un máximo de 49 esquinas a detectar para un tablero regular de ajedrez. Los restantes puntos son

posteriormente generados por medio del modelo activo de forma.

Debido a que estos métodos resultan en una decisión binaria (es esquina o no lo es), se contabilizan los positivos verdaderos y los falsos negativos. La figura 3 muestra los resultados de ambos algoritmos para algunas de las imágenes. Nótese las variación de posición, escala, rotación (eje, horizontal y vertical) e iluminación de las muestras. Los positivos verdaderos alcanzados por cada algoritmo para el conjunto de imágenes se grafican en la figura 4. Adicionalmente, la figura 5 presenta la razón de verdaderos positivos (VPR) para este conjunto de imágenes, obtenida a partir de los Verdaderos Positivos y los Falsos Negativos.

Un conjunto de 20 imágenes secundario fue creado usando un tablero de ajedrez diferente, con una separación más gruesa entre cuadros como única diferencia, es decir, las líneas entre recuadros poseen un grosor mayor. Esta prueba se incluye a partir de un análisis del funcionamiento del algoritmo ChESS. De igual forma que para el primer conjunto, diferentes posiciones para el tablero fueron incorporadas. La figura 6 muestra detecciones de esquinas para este conjunto utilizando este algoritmo.

### Estimación de la pose de cámara

En la figura 7 se muestra un cubo de realidad aumentada superpuesto a algunas imágenes

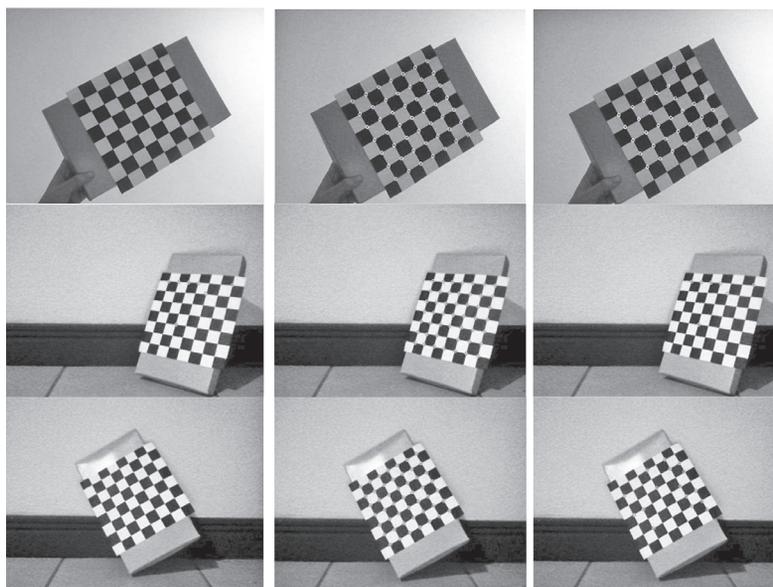


Figura 3. Detección de esquinas en tablero de ajedrez. a) Imagen original, b) Algoritmo ChESS, c) Algoritmo heurístico

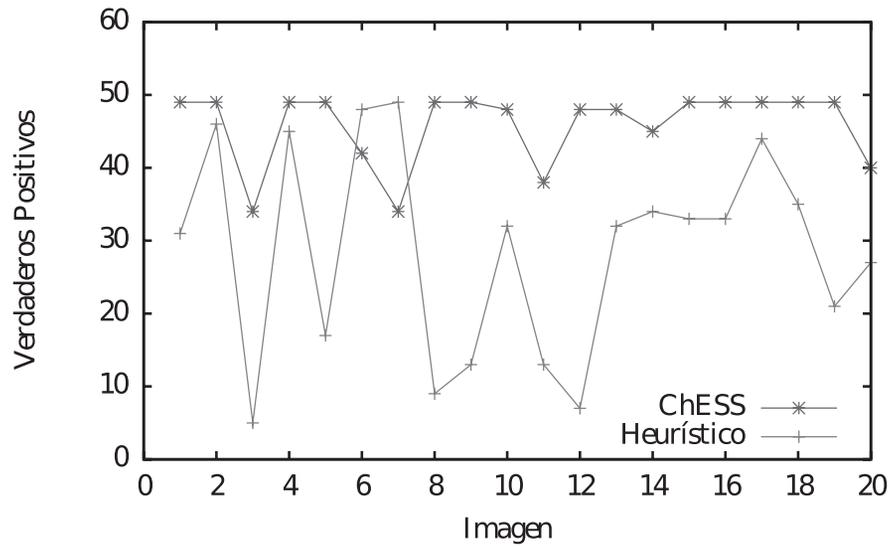


Figura 4. Positivos verdaderos alcanzados para el conjunto de imágenes usando algoritmos comparados (imágenes de 640x480 de resolución).

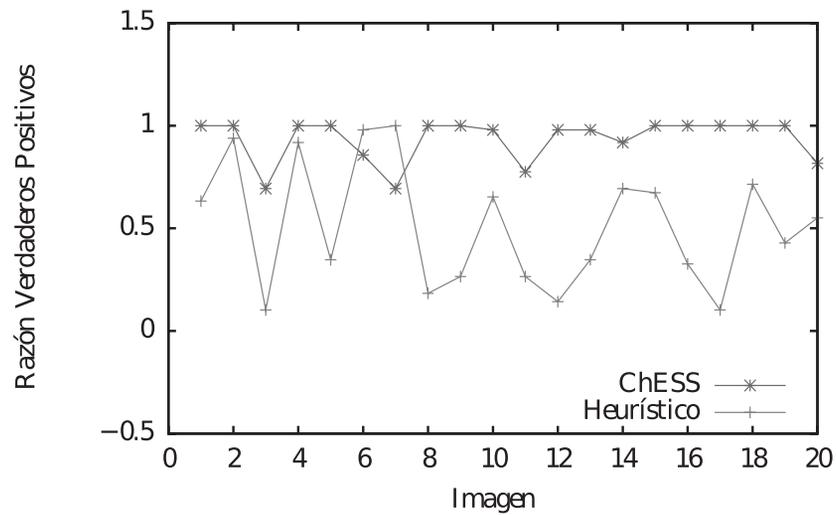


Figura 5. Razón de verdaderos positivos (VPR) para el conjunto de imágenes usando algoritmos comparados (imágenes de 640x480 de resolución).

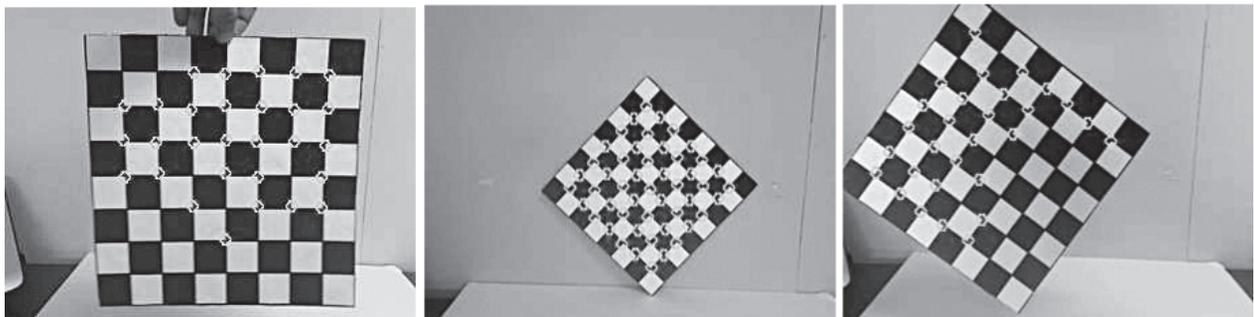


Figura 6. Detecciones de esquinas en tablero de ajedrez con borde grueso usando el algoritmo ChESS.

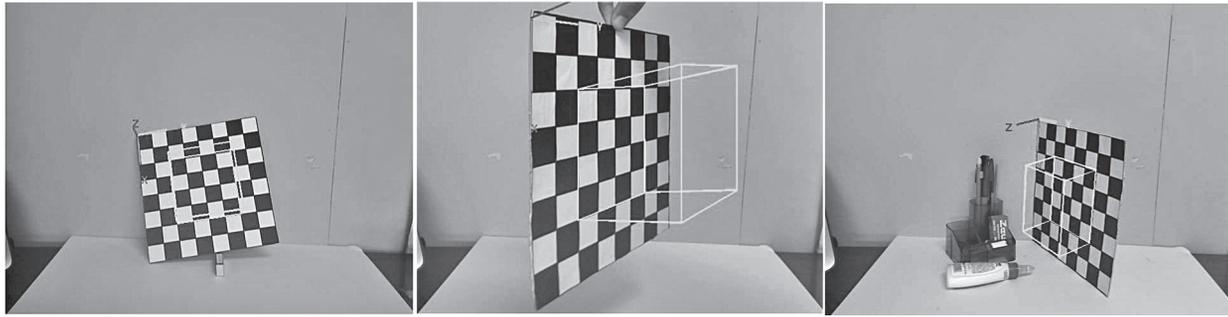


Figura 7. Realidad aumentada en tablero de ajedrez usando el método de estimación de pose propuesto.

utilizadas. Las coordenadas de las esquinas del cubo se calcularon con base en los parámetros de pose de cámara estimados con el método propuesto (Zhang, 2000).

El cuadro I presenta la variación normalizada de los parámetros de pose de cámara estimados con el método de Zhang contra los parámetros obtenidos con la herramienta Camera Calibration Toolbox de Matlab® para las tres imágenes de la figura 3.

### Discusión

Es claro a partir de los resultados de las figuras 4 y 5 que el algoritmo ChESS posee mayor exactitud en la detección de esquinas para tablero. En promedio se obtuvo una razón de verdaderos positivos para la detección del 93,46%, es decir, el algoritmo es capaz de detectar correctamente una esquina del tablero de ajedrez un 93,46% de las veces. Por otro lado, el algoritmo heurístico presenta mayor inexactitud en comparación con ChESS, con solo un 51,13% de razón de verdaderos positivos promedio para la detección de las esquinas y una variación considerable de positivos verdaderos entre escenas. Se escoge así ChESS como algoritmo de mayor confiabilidad.

Aún así, el método ChESS presenta una limitación mayor. Como se observa en la figura 6, y en comparación con los resultados de la figura 3, el grosor de las líneas de separación entre recuadros del tablero de ajedrez está directamente relacionado con la eficacia del algoritmo. Este hecho se origina en la operación del algoritmo en sí, donde el uso de un rango de comparación independiente del tamaño total de la imagen se refleja en los resultados.

Cuadro I. Variación porcentual entre parámetros estimados con algoritmos propuesto contra la herramienta de Matlab®

	Variación de parámetro (%)		
K	0.86	0	13.91
	0	-4	21.12
	0	0	0
$R_1$	18.4	-20.82	14.09
	-3.66	9.64	-50.07
	-79.46	-18.85	-11.81
$t_1$	-94.05	-115.09	-96.98
$R_2$	11.42	-25.17	-490.64
	-5.67	3.07	-39.63
	-0.67	-72.72	-26.93
$t_2$	-109.34	-102.58	-96.91
$R_3$	39.65	-13.43	-1.31
	6.15	17.43	-16.85
	-10.89	16.14	9.14
$t_3$	-103.13	-99.53	-96.80

La alta variación entre vectores de traslación  $t_1$ ,  $t_2$  y  $t_3$  es explicada como una consecuencia de un desajuste de unidades. Las unidades de medición utilizadas en ambos métodos difieren y, por condiciones inherentes a cada implementación, no pueden ser modificadas. Otra prueba de este desajuste se evidencia en los resultados para las matrices de rotación  $R_1$ ,  $R_2$  y  $R_3$  donde el desajuste no afecta las mediciones pues las rotaciones son adimensionales. Una vez que se ajustan las unidades luego de la obtención de resultados los valores mejoran. Nótese que en general, aunque se dan variaciones de hasta 79%, éstas se basan en píxeles y tendrían un

error máximo de 6 píxeles. Esto se observa además en la Figura 7, donde el cubo de realidad aumentada generado a partir de estos parámetros se acopla a la posición del tablero con leves problemas de ajuste.

En términos generales, los resultados obtenidos por medio de la implementación del método de Zhang para la estimación de pose de cámara varían considerablemente de los resultados dados por la herramienta Matlab®. Esto posiblemente es causado por ruido en la detección de esquinas debido a que el método de extracción del patrón debe de ser muy exacto, de lo contrario los resultados finales no serán confiables, como se evidenció. Además, en esta implementación, por motivos de aplicación final orientada a un sistema embebido, solo se utilizaron cuatro puntos de interés por imagen para la estimación de pose. Un análisis más profundo del algoritmo utilizado por la herramienta Matlab® reveló que éste utiliza todo el conjunto de puntos de interés detectados. Esta exactitud, sin embargo, consume potencia computacional que solo puede alcanzarse con alto rendimiento en una computadora de propósito personal.

## Conclusiones y trabajo futuro

La evaluación del algoritmo ChESS contra el método heurístico probó que, mientras se utilice un tablero de ajedrez adecuado al funcionamiento del algoritmo ChESS, este método presenta un mejor rendimiento independientemente de la posición y rotación del tablero, con una detección promedio de 93,46%.

Relativo a la estimación de pose de la cámara, la comparación entre el algoritmo implementado y la herramienta de calibración de Matlab® comprobó que con una mayor cantidad de puntos de interés utilizados en el algoritmo de estimación de pose se eleva la eficiencia de los resultados, sin embargo, disminuye considerablemente el rendimiento en términos de consumo computacional, aspecto no apto para la implementación en un sistema embebido.

Para versiones futuras debe realizarse un balance entre eficiencia de detección y rendimiento computacional utilizado, pues se deben considerar las limitaciones de memoria inherentes a un sistema embebido. Se puede aplicar una optimización para el algoritmo ChESS a fin de mejorar la exactitud de la detección de los puntos de interés.

## Bibliografía

- Alvarado, P. (2012, mayo 4). LTI-LIB2. A *Computer Vision Library*. Disponible en: <http://www.ie.itcr.ac.cr/palvarado/ltilib-2/html/main.html>
- Bennett, S. & Lasenby, J. (2013). *ChESS - Quick and Robust Detection of Chessboard Features*. ArXiv e-prints Journal.
- Cootes, T.F., Taylor, C.J., Cooper, D.H. & Graham, J. (1995). Active Shape Models. Their Training and Application. *Computer Vision and Image Understanding* 61(1): 38-59.
- González, R. & Woods, R. (2008). *Digital Image Processing*. Prentice Hall.
- Kassir, A. & Peynot, T. (2010). *Reliable Automatic Camera-Laser Calibration*. Proceedings of the 2010 Australasian Conference on Robotics & Automation ACRA 2010, Australia.
- Madsen, K., Nielsen, H.B. & Tingleff, O. (2004). *Methods for non-linear least squares problems*. Informatics and Mathematical Modelling, Technical University of Denmark (DTU).
- Mittrapiyanuruk, P. (2006). *A Memo on How to Use the Levenberg-Marquardt Algorithm for Refining Camera Calibration Parameters*. Robot Vision Laboratory, Purdue University, West Lafayette, IN, USA.
- Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Microsoft Research.
- Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(11): 1330- 1334.

## Apéndice (opcional)

Para visualizar un ejemplo del sistema en funcionamiento para video, visitar el enlace: <http://www.youtube.com/watch?v=rRbaiEy3B0Y>