

# ChessTrack

## Seguimiento de una partida de Ajedrez.

Nicolás Furquez - Aylén Ricca

### Descripción del problema.

El problema planteado se presenta en el marco de una partida de ajedrez clásica, en donde interesa transcribir a notación algebraica<sup>1</sup> la partida en cuestión, en tiempo real, para ser enviada a un servidor o un jugador remoto. Como hoy en día es muy accesible tener un celular con la potencia para de procesar imágenes, el proyecto será enmarcado dentro de la tecnología Android<sup>2</sup>.

Como antecedentes se cuenta con un proyecto de la asignatura llevado a cabo en 2013, 8x8CAM, el cual es similar en características pero desarrollado para PC y sólo se tomaban fotogramas del tablero antes y después del movimiento, en nuestro caso el procesamiento es en tiempo real.

El tablero utilizado será uno profesional como los de la *figura 1*, proporcionados por el tutor; se utilizará un ambiente donde la iluminación será controlada y la partida será "lenta", con esto se quiere decir que se dará tiempo al dispositivo a procesar la partida entre movimiento y movimiento.



**FIGURA 1.** Tableros a utilizar durante el desarrollo.

Como objetivo opcional, se plantea la posibilidad de transcribir la partida a voz para jugadores no videntes.

Dado que una partida normal de ajedrez es muy rica en movimientos<sup>3</sup>, en un principio

<sup>1</sup> [http://es.wikipedia.org/wiki/Notaci%C3%B3n\\_algebraica](http://es.wikipedia.org/wiki/Notaci%C3%B3n_algebraica)

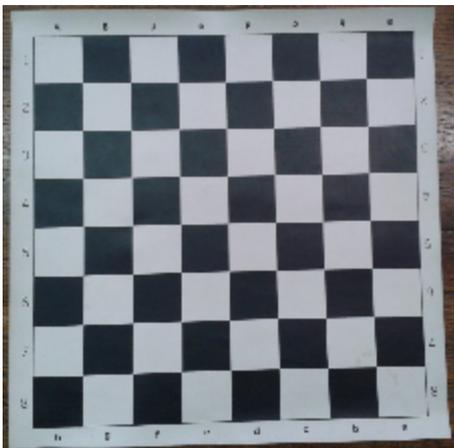
<sup>2</sup> <http://developer.android.com/index.html>

<sup>3</sup> [http://es.wikipedia.org/wiki/Reglamento\\_del\\_ajedrez](http://es.wikipedia.org/wiki/Reglamento_del_ajedrez)

se plantea la detección de una partida con movimientos básicos, en la segunda iteración se incluirá la detección de jake y jake mate, valiéndose de las reglas oficiales de ajedrez, si el tiempo lo permite se tratará de detectar enroques, coronaciones (promociones) y captura al paso, igualmente se tendrá en cuenta en el diseño de la arquitectura estos casos especiales para una futura extensión.

Para procesar las capturas de una forma más eficiente se usará la biblioteca *OpenCV4Android*<sup>4</sup> dado que cuenta con una gran cantidad de utilitarios para el procesamiento de imágenes.

### Imágenes y Videos de ejemplo.



**IMAGEN 1.** Tablero vacío.



**IMAGEN 2.** Tablero inicial.



**IMAGEN 3.** Tablero con oclusión por mano.



**IMAGEN 4.** Partida iniciada.



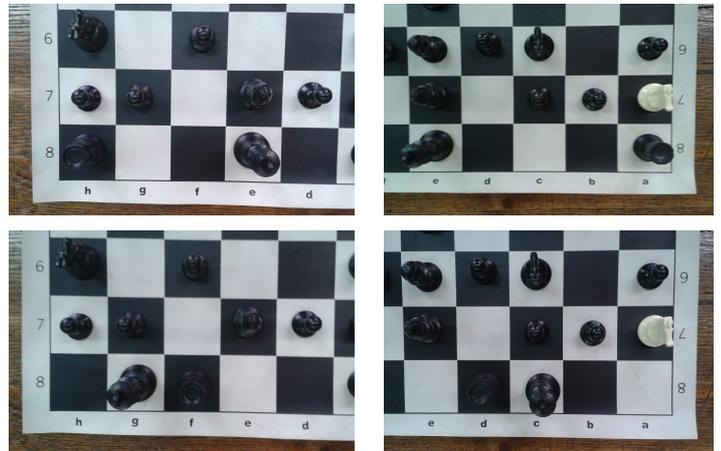
**IMAGEN 5.** Jaque sobre rey blanco (izq) - Jaque sobre rey negro (der).

---

<sup>4</sup> [http://docs.opencv.org/doc/tutorials/introduction/android\\_binary\\_package/O4A\\_SDK.html](http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/O4A_SDK.html)



**IMAGEN 5.** Jaque mate.



**IMAGEN 6.** Enroque corto (antes y después) - Enroque largo (antes y después).



1. e4 e5
2. f4 exf4
3. Ac4 Dh4+
4. Rf1 b5
5. Axb5 Cf6
6. Cf3 Dh6
7. d3 Ch5
8. Ch4 Dg5
9. Cf5 c6
10. g4 Cf6
11. Tg1 cxb5
12. h4 Dg6
13. h5 Dg5
14. Df3 Cg8
15. Axf4 Df6
16. Cc3 Ac5
17. Cd5 Dxb2
18. Ad6 Axc1
19. e5 Dxa1+
20. Re2 Ca6
21. Cxg7+ Rd8
22. Df6+ Cxf6
23. Ae7++

**VIDEO.** Reproducción de la partida Inmortal (Londres, 1851). (<https://www.youtube.com/watch?v=GdQmgzBK0Jo>)

## Técnicas a emplear.

Al iniciar la adquisición, se realizará una ecualización de histograma que buscará realzar el contraste de los elementos presentes en la imagen sin pérdida de información, esto facilitará la identificación de los diferentes elementos en las capturas.

En primer lugar se desea detectar el tablero, para lo cual se probarán técnicas de detección de rectas y/o esquinas esperando encontrar resultados positivos. La idea inicial es poder llevar el problema a una dimensión igual a 8x8 posiciones (casilleros del tablero) en donde ocurren novedades entre frames capturados.

Se pretende utilizar una transformación homográfica para enderezar el tablero si el mismo no está correctamente alineado, para ello será necesario contar con las cuatro

esquinas externas del tablero correctamente identificadas.

Para mantener el estado del tablero en todo momento, es decir, la ubicación de cada una de las 8x2 fichas en el tablero, no se buscará reconocer las distintas piezas (dama, caballo, peón, etc.) puesto que al tener una visión superior del tablero, la diferenciación de piezas se torna complicada; se atacará entonces este problema teniendo como información la ubicación inicial de todas las fichas según las reglas del juego y actualizando su posición con cada cambio del tablero.

Se segmentarán los distintos elementos de interés, como por ejemplo las fichas, y así obtener su ubicación en el tablero.

Para detectar novedades sobre el tablero aparecen dos problemas a resolver; por un lado detectar que ocurre un movimiento producto de una jugada, y actualizar el estado de las piezas involucradas; y por otro lado detectar que se está llevando a cabo un movimiento evidenciado por la oclusión del tablero por parte de los jugadores. Para resolver el primer problema planteado, se realizará la doble diferencia entre capturas, identificando qué posición se vació respecto a la primer captura y que nueva casilla está ocupada en la segunda captura que no lo estaba en la primera. Respecto a la solución del segundo problema, se investigarán técnicas de detección de piel u otra similar para saber cuándo un jugador está realizando una movida.

## Cronograma.

### ChessTrack

Start: junio 29, 2014

Finish: julio 23, 2014

Report Date: junio 27, 2014

### Gantt Chart

WBS	Name	Work	Week 27, 2014							Week 28, 2014							Week 29, 2014							Week 30, 2014													
			30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30				
1	Definición de Requerimientos		█																																		
2	Diseño y validación de la Arquitectura	5d	█																																		
3	Prototipo reutilizable	15d								█																											
4	Pruebas de técnicas de imágenes	10d	█																																		
5	Desarrollo de la solución	15d								█																											
6	Testing del sistema	1d															█																				
7	Documentación	1d															█																				
8	Fin del proyecto																																				

### Tasks

WBS	Name	Start	Finish	Work	Complete
1	Definición de Requerimientos	Jun 30	Jul 1		0%
2	Diseño y validación de la Arquitectura	Jun 30	Jul 4	5d	0%
3	Prototipo reutilizable	Jul 2	Jul 16	15d	0%
4	Pruebas de técnicas de imágenes	Jun 30	Jul 9	10d	0%
5	Desarrollo de la solución	Jul 6	Jul 20	15d	0%
6	Testing del sistema	Jul 10	Jul 21	1d	0%
7	Documentación	Jul 3	Jul 23	1d	0%
8	Fin del proyecto	Jul 24	Jul 24		

## Bibliografía.

- ❑ Proyecto 8x8CAM realizado en 2013 de similares características - <http://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2013/ajedrez/index.htm>
- ❑ Chavarro, PhD Julio C. "Ajedrez con accesibilidad y autonomía para personas en situación de discapacidad visual."
- ❑ G.D. Illeperuma "Using Image Processing Techniques to Automate Chess Game Recording."
- ❑ Arda Mustafa, Kuşçu Hilmi, Öztürk Emir "Obtain Chess Moves with Image Processing."
- ❑ Fédération Internationale des Échecs (Federación Internacional del Ajedrez) <http://www.fide.com/>
- ❑ Biblioteca OpenCV - <http://opencv.org/>
- ❑ Android Development - <http://developer.android.com/index.html>
- ❑ OpenCV4Android - [http://docs.opencv.org/doc/tutorials/introduction/android\\_binary\\_package/O4A\\_SDK.html](http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/O4A_SDK.html)