

Checkmate!

Paul Ardis

Tom Scogland

Andy Zeigler



Overview

- 1) Introduction
- 2) Goals
- 3) Prior Work
- 4) Implementation
- 5) Challenges



Overview

- 1) Introduction
- 2) Goals
- 3) Prior Work
- 4) Implementation
- 5) Challenges



Introduction

- Design an autonomous and portable mixed-reality system that enables a computer to play games against a human
- Will use standard chessboard and pieces



Overview

- 1) Introduction
- 2) **Goals**
- 3) Prior Work
- 4) Challenges
- 5) Implementation



Goals

- Goals
 - Design recognition engine
 - Focus our efforts on recognition and arbitration
 - Build simple AI



Overview

- 1) Introduction
- 2) Goals
- 3) **Prior Work**
- 4) Implementation
- 5) Challenges



Prior Work

- 2002, Ecole Polytechnique
 - 3 students design autonomous chess playing robot with vision and a robotic arm
- Jamca
 - A small, simple, open-source chess board implementation



Prior Work

- Internet chess community
 - Many, many playing engines and board display programs
 - Most are open and/or interoperable via existing standards
 - We aim for standards compliance!!



Overview

- 1) Introduction
- 2) Goals
- 3) Prior Work
- 4) Implementation
 - a) Representation
 - b) Calibration
 - c) Gameplay
- 5) Challenges



Representation

- Checkmate “knows” how to play chess
- Board is represented in internal structures
- Implements all chess rules
 - 50-move rule
 - En-passant
 - Castling



The Board

110	111	112	113	114	115	116	117	118	119
100	101	102	103	104	105	106	107	108	109
90	91	92	93	94	95	96	97	98	99
80	81	82	83	84	85	86	87	88	89
70	71	72	73	74	75	76	77	78	79
60	61	62	63	64	65	66	67	68	69
50	51	52	53	54	55	56	57	58	59
40	41	42	43	44	45	46	47	48	49
30	31	32	33	34	35	36	37	38	39
20	21	22	23	24	25	26	27	28	29
10	11	12	13	14	15	16	17	18	19
0	1	2	3	4	5	6	7	8	9



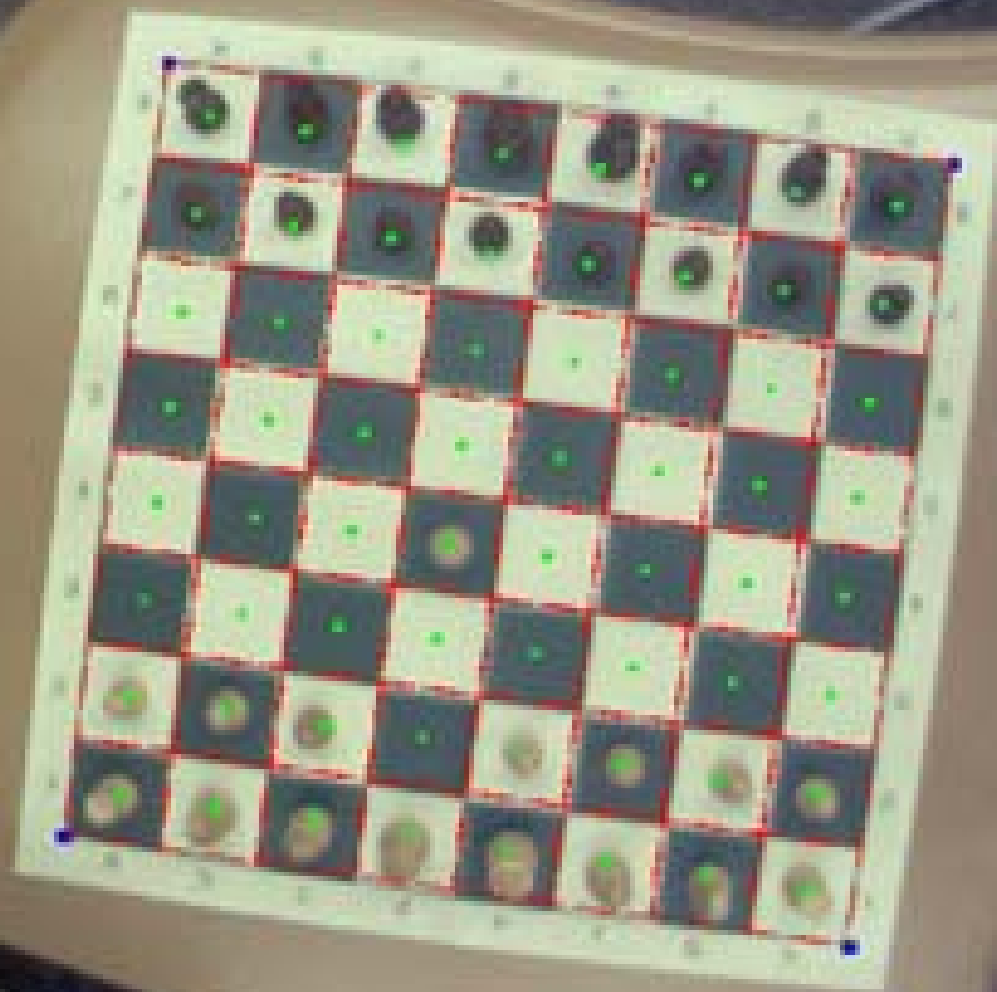
Calibration

- How does it “see” the chess board?
 - Auto-calibrate camera for specific lighting conditions
 - Gamma, color balance, etc.
 - Calibrate board
 - User clicks on corners
 - Checkmate knows board geometry
 - Using corners of board, it unprojects board plane to an orthogonal view









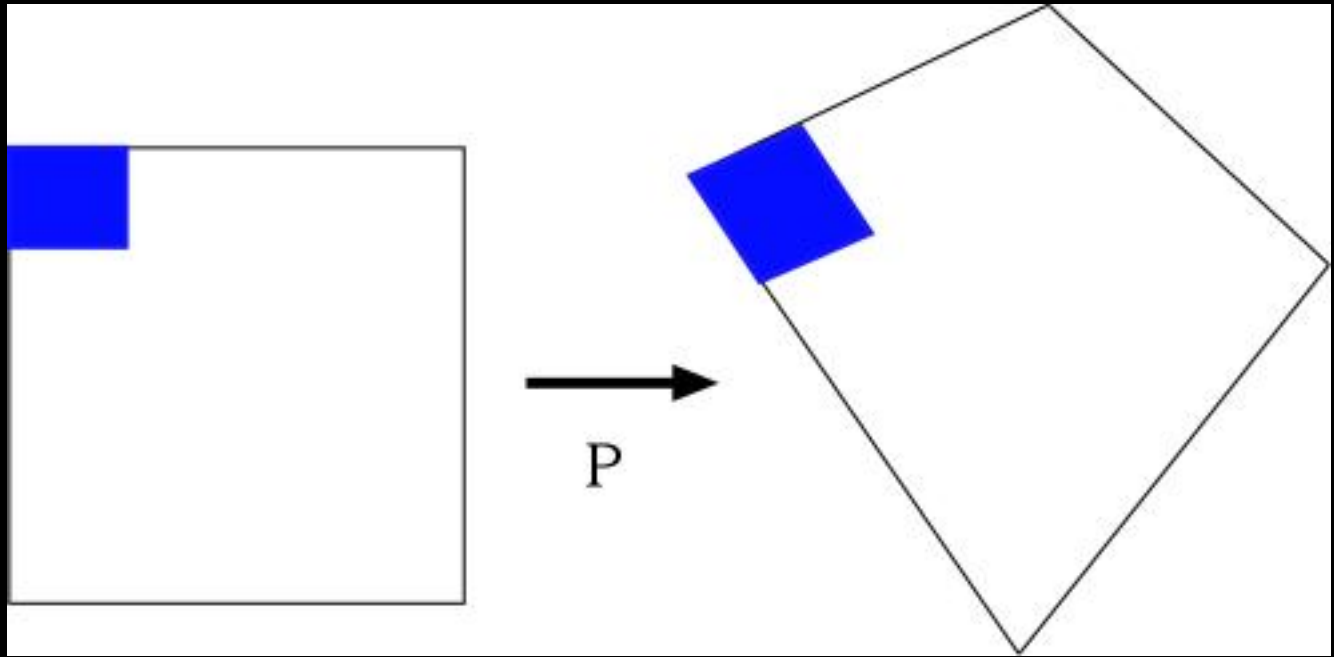
Gameplay

- Stores camera shot
- Calculates perspective matrix
- Divides unit square into regions
- Maps camera regions to unit regions
- Determines changes to centers of squares
- Finds most-changed



Gameplay (cont.)

Move Finding



Unit Square

Actual Image











Gameplay (cont.)

- Artificial Intelligence
 - Input: List of legal moves
 - Output: A “good” move
 - Point system (10, 5, 3, 3, 1)



Gameplay cont.

- Artificial Intelligence (cont.)
 - Classification of moves
 - Capture
 - Risk
 - Evasion
 - Want to capture safely if possible



Implementation

- Demo time
 - Play a game of chess



Overview

- 1) Introduction
- 2) Goals
- 3) Prior Work
- 4) Implementation
- 5) Challenges



Challenges

- Accurately handling real-world piece movement
 - Black VS white pieces
 - Nonstandard board/piece colors
 - Abnomal lighting, off-color or off-center
 - Getting a reliable camera angle to avoid occlusion



Future Work

- Smooth and natural autonomous interaction
 - Showing which piece the computer wants to move, and to where
 - Explaining what is wrong with an illegal move that has been made by the player



Discussion

- Q/A

