Geometry-Based Populated Chessboard Recognition

William Hoff

whoff@mines.edu
Colorado School of Mines
Golden, Colorado, USA

bill.hoff@daqri.com
DAQRI
Vienna, Austria

My co-authors:
Youye Xie, Gongguo Tang
Colorado School of Mines
Chessboard Recognition

- Recognizing board useful for camera calibration
Chessboard Recognition

- Recognizing board useful for camera calibration
- But we are not addressing this problem!
- Want to do recognition for chess games
- Finding the board is the first step in finding the pieces
Populated Chessboard Recognition

- Challenges
  - Occlusion
  - Lighting
  - Clutter

Methods developed for camera calibration do not work on populated chessboards!

Result from OpenCV’s “findChessboardCorners”
Approach

- **Main ideas**
  - Look for two groups of lines
  - Find correspondence by fitting observed lines to the model of a chessboard

- **Key techniques**
  - Canny edge detector
  - Hough transform
  - K-means clustering
  - Homography transformation
Line Detection

• Canny edge detection + Hough transform
• Edges vote for lines into a Hough space, indexed by $\rho$ and $\theta$
• Lines are represented using the equation
  \[ \rho = x \cos \theta + y \sin \theta \]

Size of Hough space:
Width = 180 ($\theta \in \{-90^\circ..+89^\circ\}$
Height = $2d$, where $d$ is image diagonal
Line Clustering

- Find two groups of lines
  - Corresponding to the two sets of lines on a chessboard

- Use k-means clustering in H-space

- Need to scale the H-space
  - So that distance between two adjacent lines in the $\rho$ direction and the $\theta$ direction are weighted equally
Line Clustering

Image space
4030 x 3024

Hough space
5040 x 180

Scaled Hough space
180 x 180
(weight \( \rho \) and \( \theta \) equally)

Two clusters
Outlier Elimination

- Eliminate lines that are far from the other lines in its own group, in H-space

Avg. Euclidean distance within sets:
- Green: 3.3107
- Red: 3.8320

Avg. Euclidean distance between sets:
- 64.9462

< 3x average distance
Outlier Elimination (image space)

Before

After (with intersections)
Finding Correspondence

- Model of chessboard: 81 intersection points

- Find transformation that maps observed intersections to model intersections

- Use a homography (projective transform)
  - Maps image points to model points

\[
\begin{pmatrix}
    x_1 \\
    x_2 \\
    x_3
\end{pmatrix} =
\begin{pmatrix}
    h_{11} & h_{12} & h_{13} \\
    h_{21} & h_{22} & h_{23} \\
    h_{31} & h_{32} & h_{33}
\end{pmatrix}
\begin{pmatrix}
    x \\
    y \\
    1
\end{pmatrix}
\]
Finding Correspondence (continued)

- Compute transform from four corner points and count inliers
- Search from outer corners to inner corners
Experiments - Conditions

• Occlusion
  1. No pieces on board
  2. A chess middle game
  3. All pieces in initial positions

• Viewing angles
  – [10, 20, …, 90] degrees

• Testing
  – 4 different chess sets
  – 20 to 30 test images for each combination
Example Results
Processing Time

- Processing time dominated by correspondence search
  - Search time depends on the number of lines to be searched

- Using the top 35 lines, processing time is about 4 seconds
- MATLAB implementation on i7 processor
Detection Success Rate

No pieces on board

Test image example
Detection Success Rate

Chess middle game

Test image example
Detection Success Rate

Pieces in initial positions

Test image example
Failure Cases

- Very low viewing angle
- Strong occlusion
- Strong spurious edges
Piece Recognition

- Once the board is recognized, we can use a model-based method to recognize the pieces
- We use chamfer matching to match observed edges to predicted edges in a template
Example Results
Piece Recognition Evaluation

Confusion matrix
~94% accuracy
Conclusions

• Developed algorithm for populated chessboard recognition
  – Detect lines using Hough transform
  – Cluster lines in H-space
  – Find correspondence using model-based projection
• Achieves high accuracy for >45° viewing angle
• Takes about 4 seconds for a high resolution (4032x3042 pixels) image
• Result can enable piece recognition
Thank you!