Validation methods for geometric camera calibration

Paul Romanczyk

paul@imatest.com

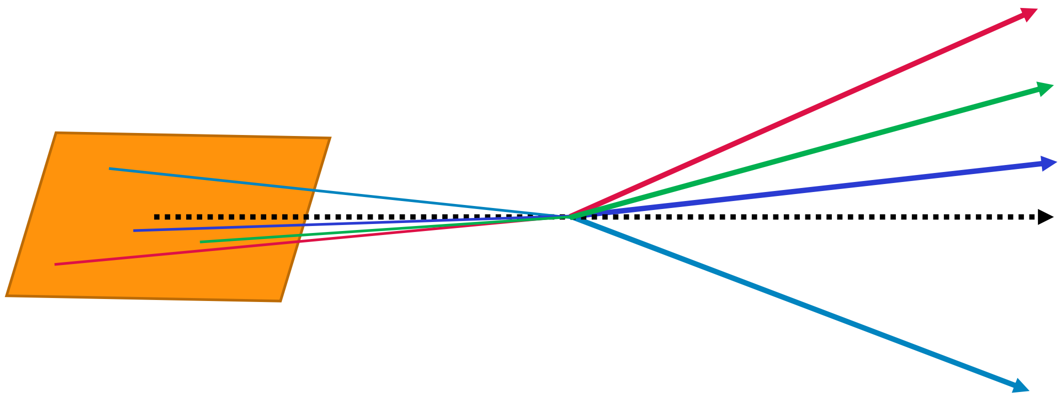
2020-01-28

Outline

- (Brief) introduction to geometric calibration
- Image coordinate validation
- Monocam reprojection validation (image formation direction)
- Monocam projection validation (point projection direction)
- Stereo-pair triangulation validation

Geometric Calibration

- A mapping of pixel locations to direction in the world
  - Mathematical model
  - May have correspondence with physical camera properties



Why Calibrate?

- Overcome manufacturing tolerance limitations to extract more performance out cameras
  - Looser manufacturing tolerances allow for lower cost/higher yield
- A geometric calibration may be used to transform pixel-based metrics to physically based ones
  - E.g., world projected SFR, IFOV measurements for photo- or radio-metric calculations

Why Validate a Calibration?

- Ensure a camera's calibration meets the geometric requirements from an ADAS system
- Compare different calibration models and/or parameterizations

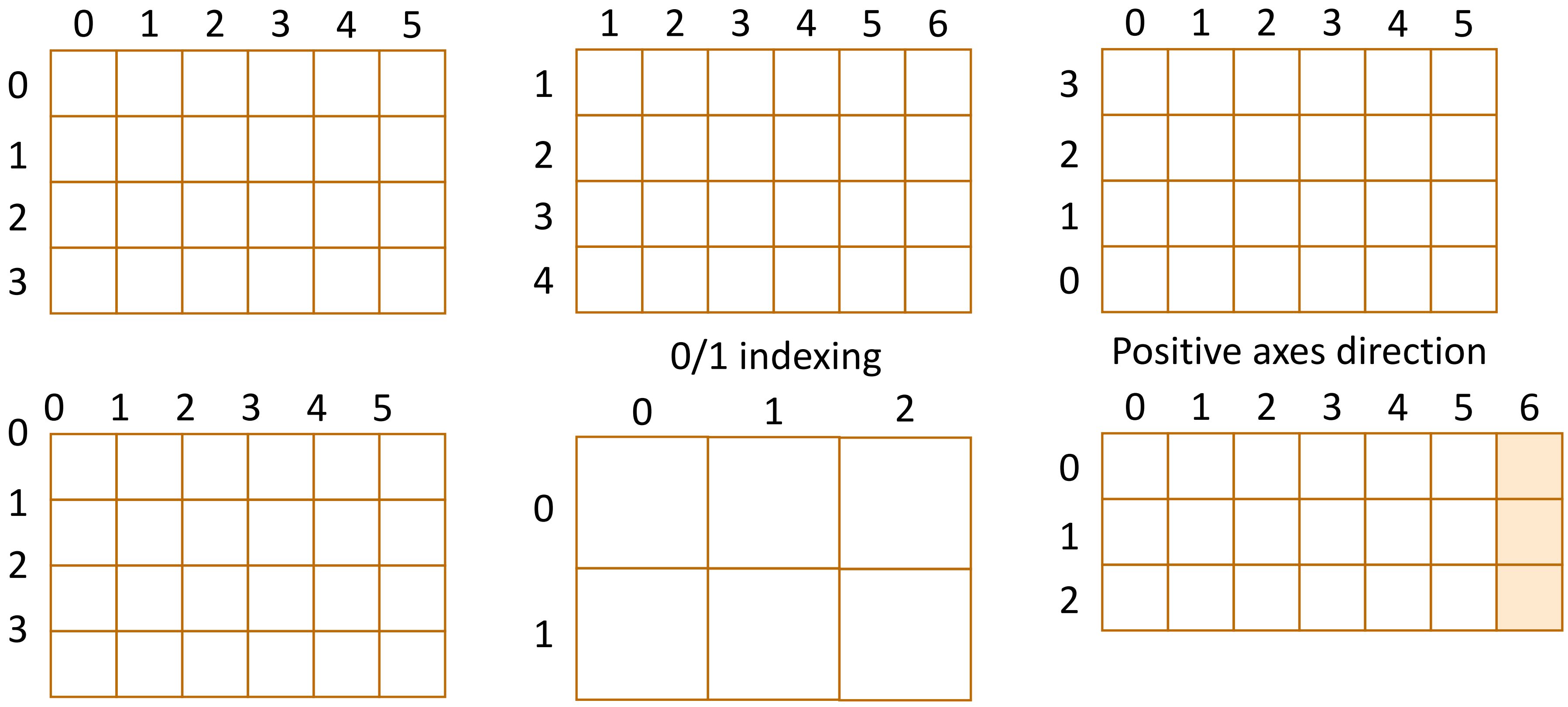
- Hard to standardize the calibration process
  - Too many models (polynomial, division polynomial, Brown-Conrady, Kannala-Brandt, custom), camera types (monocams, multicams, narrow-FOV, wide-FOV), use cases (forward-facing, back-up, in-cabin, mirror-replacement)
  - More will be coming

Desired Properties of a Validation Method

- Indicative of goodness of calibration
- Camera model agnostic
- Standardizable

Coordinate System Validation

- Make sure different teams/companies are talking the same language



0/1 indexing

Positive axes direction

Reference location

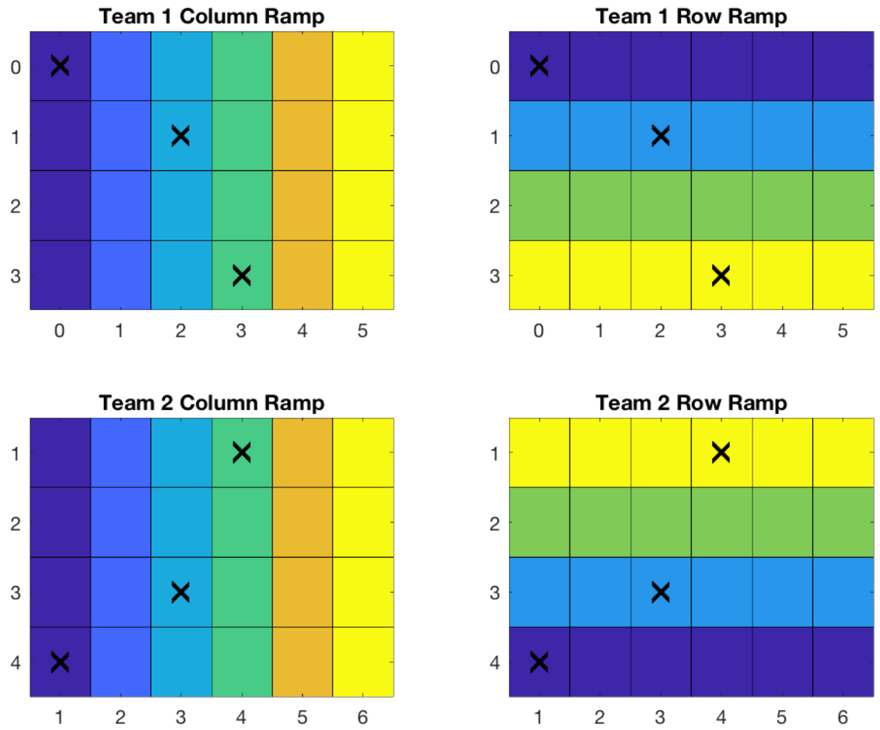
Up/down-sampling

Cropping/appending data

Image Coordinate System Validation

1. Generate a row-index and column-index sensor-test pattern from the sensor through a capture pipeline
   - This is intended on being an unambiguous reference coordinate system
   - Requires access to sensor test patterns, if they exist
2. Agree on at least 3 non-row and non-column colinear reference points in the test-pattern coordinate system
3. Find the location of the reference coordinates in your coordinate system
4. Perform an ordinary least squares regression on both the row and column coordinates

Image Coordinate System Validation Example



Sensor x	Sensor y	Team 1 x	Team 1 y	Team 2 x	Team 2 y
2	1	2	1	3	3
3	3	3	3	4	1
0	0	0	0	1	4

	$x_1 = x_s$ $y_1 = y_s$	$x_2 = x_s + 1$ $y_2 = 4 - y_s$
$x_s = x_1$ $y_s = y_1$		$x_2 = x_1 + 1$ $y_2 = 4 - y_1$
$x_s = x_2 - 1$ $y_s = y_2 + 4$	$x_1 = x_2 - 1$ $y_1 = y_2 + 4$	

Monocam Reprojection Test

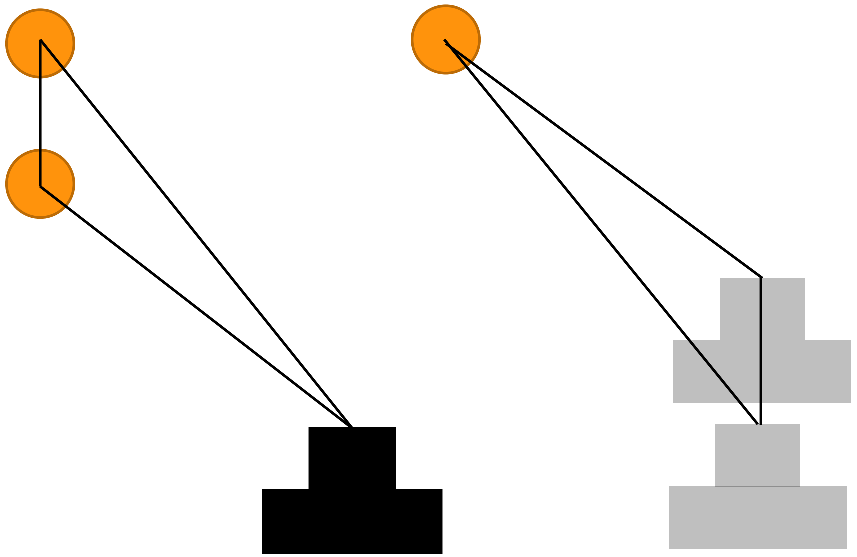
1. Create test setup with targets in the scene
2. Measure the location of the targets relative to the camera's coordinate system (pose)
3. Take an image of the target(s)
4. Perform detection of the target(s) within the image
5. Project (image formation direction) the target location through the camera model
6. Compute the error between the detected and projected points in image space (reprojection error)

Monocam Projection Test

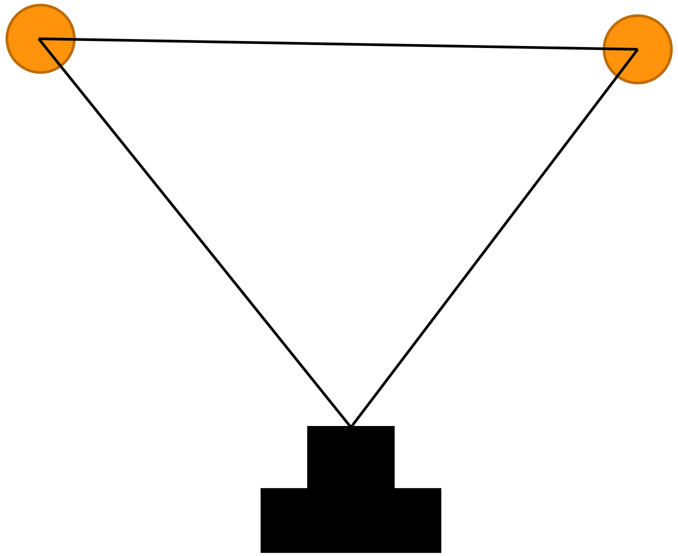
1. Create test setup with target pairs in the scene
2. Measure the distance from the camera to each target
3. Measure the distance between the target pair(s)
4. Take an image of the target pair(s)
5. Perform detection of the target pair(s) within the image
6. Project each detected image point out into the world to the distance to target (step 2)
7. Compute the distance between the projected points for each pair
8. Compare the projected distance to the measured target-pair distance (step 3)

Monocam Projection Test Configuration

SLAM



Distortion

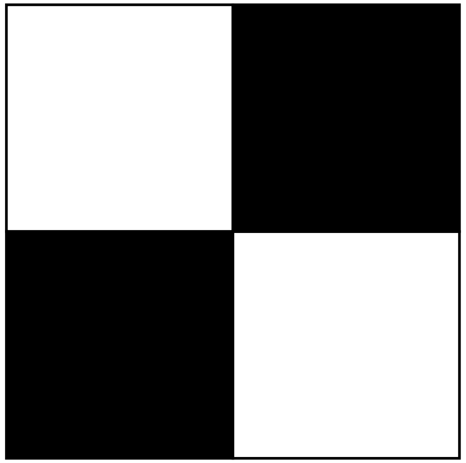
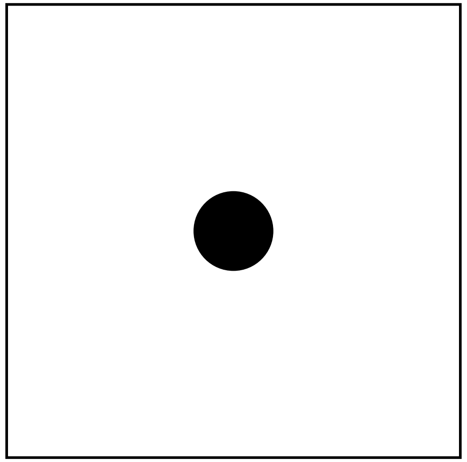


- Simulate driving forward while tracking a target
- May use multiple images moving camera

Triangulation Test

1. Create test setup with target(s) in the scene
2. Measure the location of the target(s) relative to the reference camera's coordinate system (pose)
3. Take an image of the target(s) with both cameras
4. Perform detection of the target(s) within the image from each camera
5. Perform the triangulation of the target(s) using the detected points
6. Compute the error between the measured location and the triangulated location of the target

Target Choice

	Chessboard	Dot	Automotive-Specific
			
Reference	Checker intersection	Circle center	Defined by target detection algorithm
Use	Access to image data testing	Access to image data testing	Full-system testing Black box testing

Example Target Placement Choices

- Driven by use-case
- Target distance examples
  - 2 second following distances
  - 15 second scan ahead defensive driving
  - $d = v \cdot t$
- Target Placement
  - Targets should exercise all “usable” portions of the camera’s field of view
  - Use many targets and/or rotate/translate camera for coverage

- Target separation
  - Amount car will travel in n frames
  - Size of automotive “targets”
    - Pedestrian
    - Bicycle
    - Car
    - Box Truck
- Target repetition
  - Using multiple, similarly-spaced targets allows for separation of target detection errors from calibration errors

imatest®

Thresholds and setup uncertainty

- Set test pass/fail thresholds based on requirements flow down
  - How well does camera need locate a target to allow <use case> functionality
    - Lane assist
    - Adaptive cruise control
    - Level 2, 3, 4, …
- Perform an error propagation to understand accuracies needed for test setup
  - Positions/distances
  - Ensure test setup error contributes a minimal amount to final metric

Test Comparison

	Reprojection	Projection	Triangulation
Camera Type(s)	• Monocam • Stereo Pair (each) • Multicam (each)	• Monocam • Stereo Pair (each) • Multicam (each)	• Stereo Pair
Test Coverage	• Intrinsics • Extrinsics	• Intrinsics	• Intrinsics • Extrinsics
Required knowledge	• Camera model • Camera location from housing • Camera pose • Target location(s)	• Camera model • Camera location from housing • Distance to targets • Distance between target(s)	• Camera model • Camera location from housing • Camera pose • Target location(s)
Advantages	• Often used in calibration process	• Does not require knowledge of pose	• Evaluates stereo-pair use-case
Disadvantages	• Not tied to automotive use cases	• Uncertainty is deeply coupled with camera	• Requires 2 cameras

Camera Model Comparison

- Which model is better?
  - Lower error is better
  - Need to evaluate across field-of-view
  - Test differences within the test uncertainty are “within the test margin”
- Two calibrations are “functionally equivalent” if all validations are within thresholds

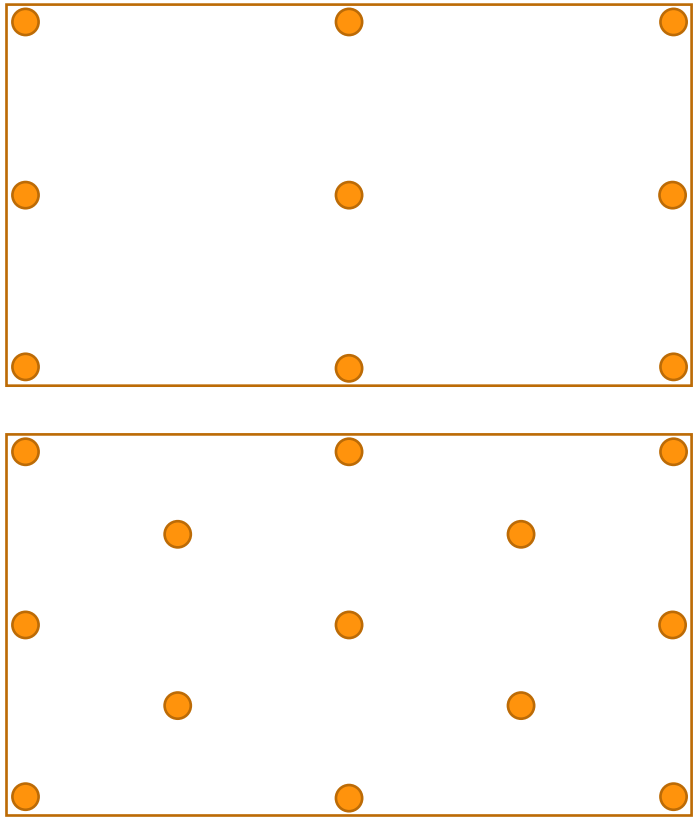
Conclusions

- Presented image coordinate validation methodology
- Presented monocam reprojection validation (image formation direction)
- Presented monocam projection validation (point production direction)
- Presented stereo-pair triangulation validation
- Tied test configurations to automotive use cases

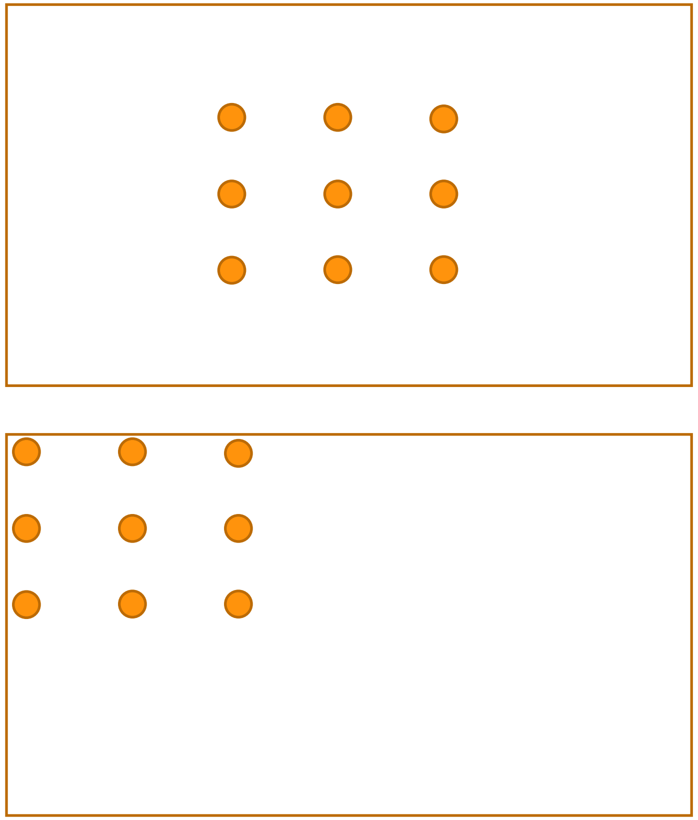
imatest®

Test Coverage

Good Coverage



Bad Coverage



Distances traveled

Velocity			Driving Time		Frame Time		
			2 [s]	15 [s]	1 [ms]	10 [ms]	100 [ms]
[m/s]	[mph]	[km/hr]	[m]	[m]	[m]	[m]	[m]
5	11.2	18.0	10	75	0.005	0.05	0.5
10	22.4	36.0	20	150	0.01	0.1	1
15	33.6	54.0	30	225	0.015	0.15	1.5
20	44.7	72.0	40	300	0.02	0.2	2
25	55.9	90.0	50	375	0.025	0.25	2.5
30	67.1	108.0	60	450	0.03	0.3	3
35	78.3	126.0	70	525	0.035	0.35	3.5
40	89.5	144.0	80	600	0.04	0.4	4
45	100.7	162.0	90	675	0.045	0.45	4.5
50	111.8	180.0	100	750	0.05	0.5	5