Single Camera Calibration
using Partially Visible Calibration Objects Based on Random Dots Marker Tracking Algorithm

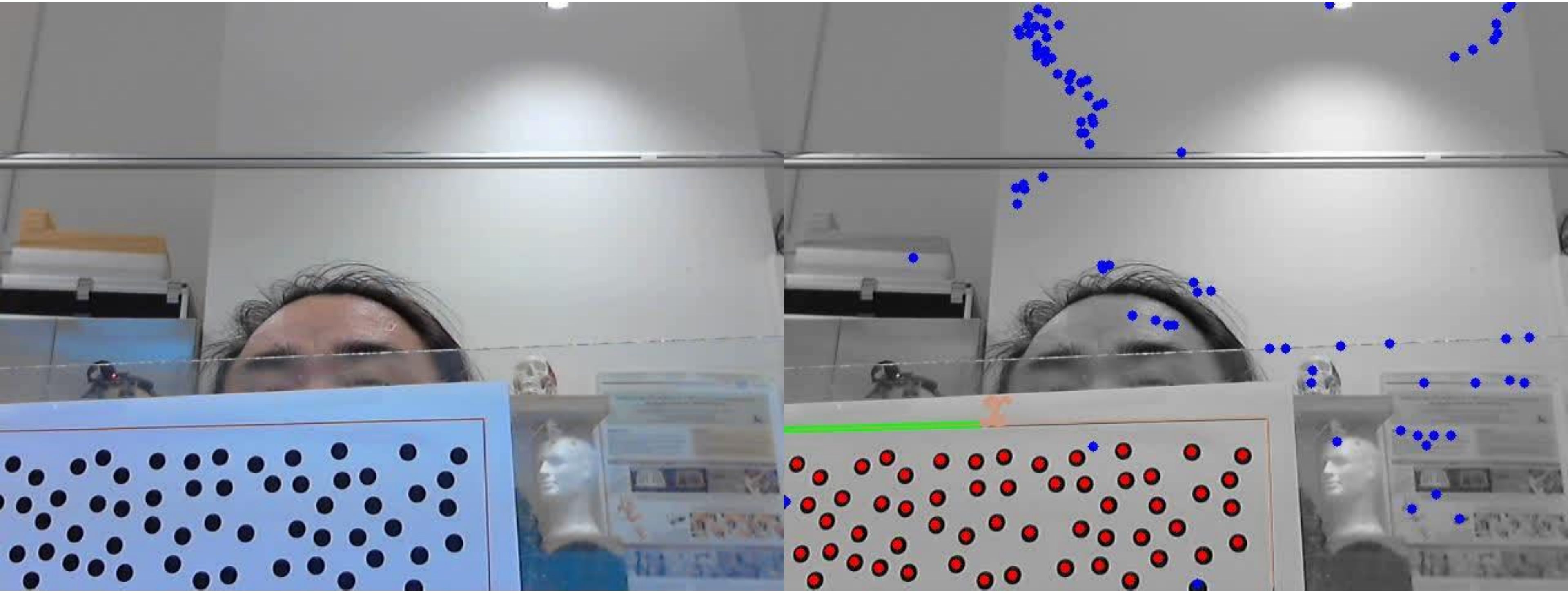
*Yuji Oyamada[1,2], Pascal Fallavollita[2], and Nassir Navab[2]

1. Keio University, Japan
2. Chair for Computer Aided Medical Procedure (CAMP), Technische Universität München, Germany

*contact: oyamada@in.tum.de
http://campar.in.tum.de/Main/YujiOyamada

Overview of this work

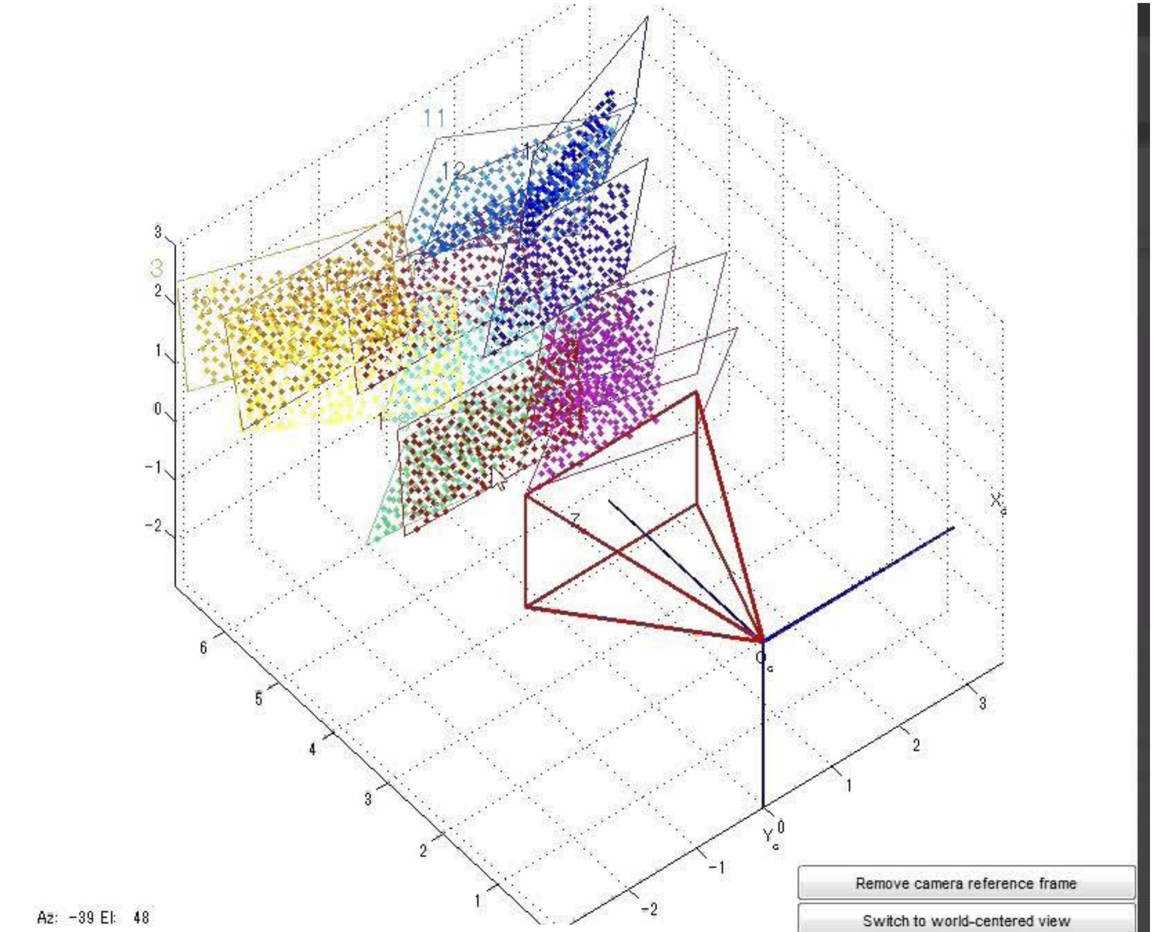
- Use a marker tracking algorithm for a single camera calibration.



Input | Marker tracking result

Overview of this work

- Use a marker tracking algorithm for a single camera calibration.

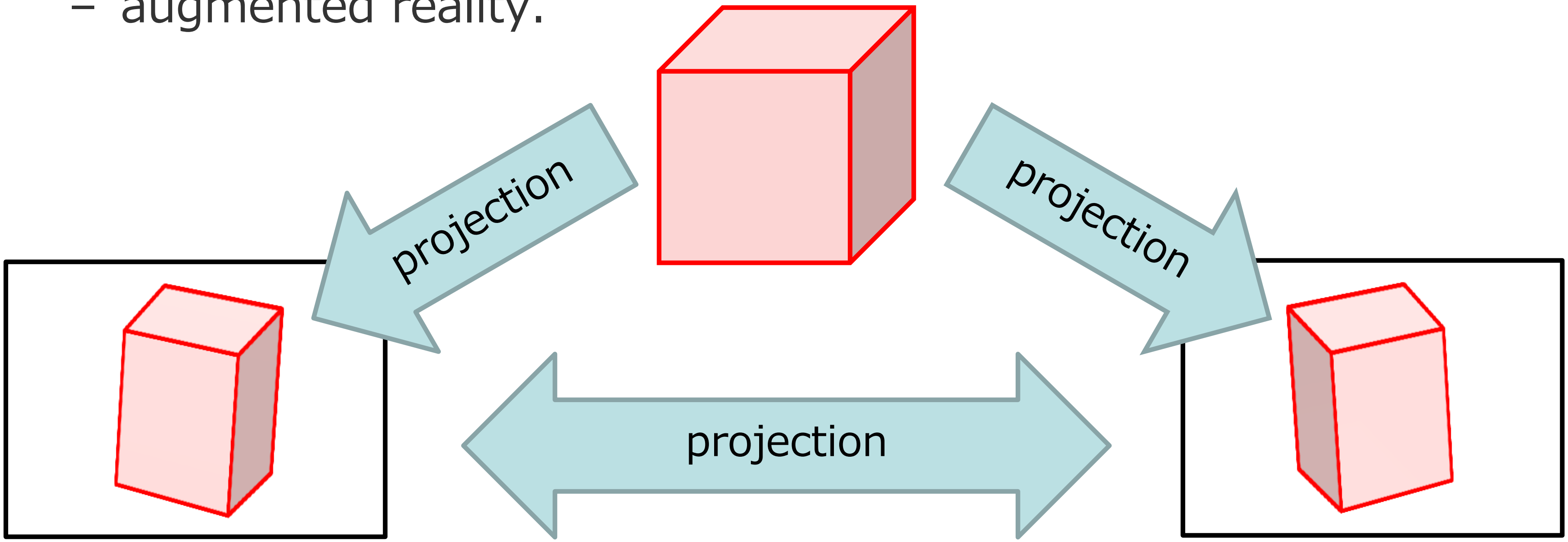


Calibration result

Introduction

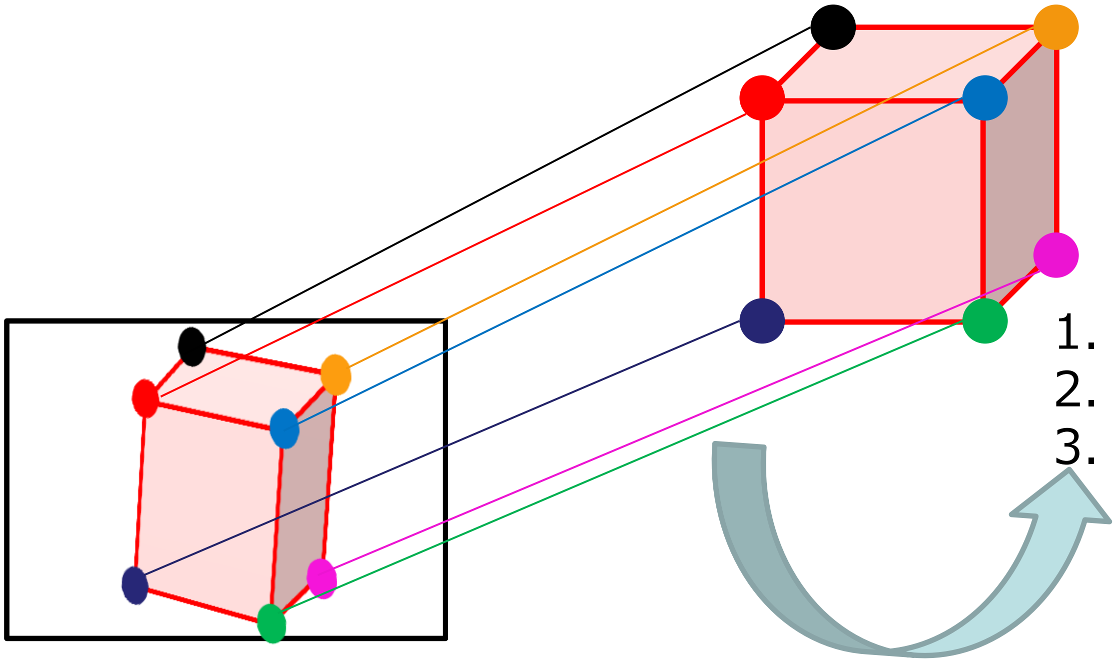
Camera calibration

- Goal: finds a relation between
  - 3D real world and 2D camera image.
  - different cameras.
- Necessary step for vision based applications:
  - 3d reconstruction,
  - augmented reality.



Calibration procedure

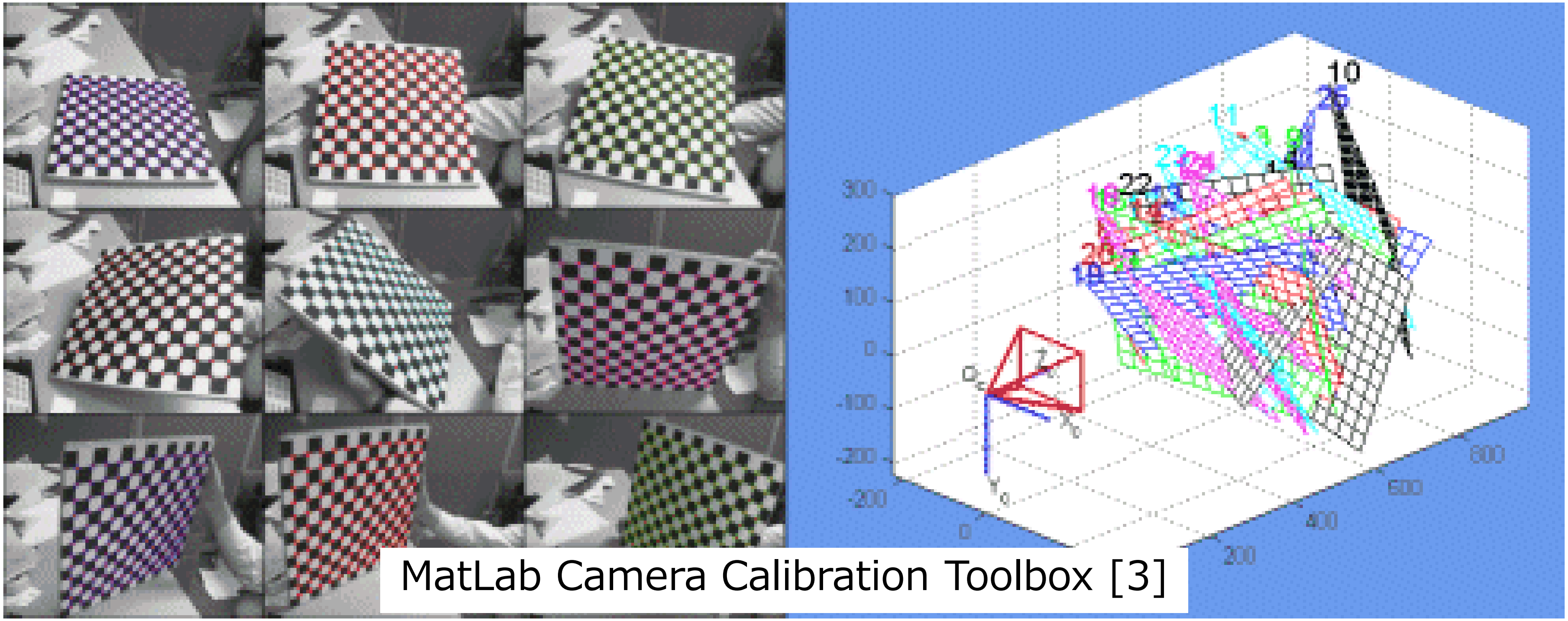
- Two main steps:
  1. finds correspondence between real world and camera images.
  2. computes parameters describing the relation.



1. Intrinsic parameters: $\boldsymbol{A}$
2. Lens distortion parameters: $\boldsymbol{\delta}$
3. Extrinsic parameters: $\boldsymbol{R}, \boldsymbol{t}$

Well-known & well-used method

- Zhang's method [26]:
  - uses several images of a set of known control points on a planar objects.
  - step-by-step parameters estimation.

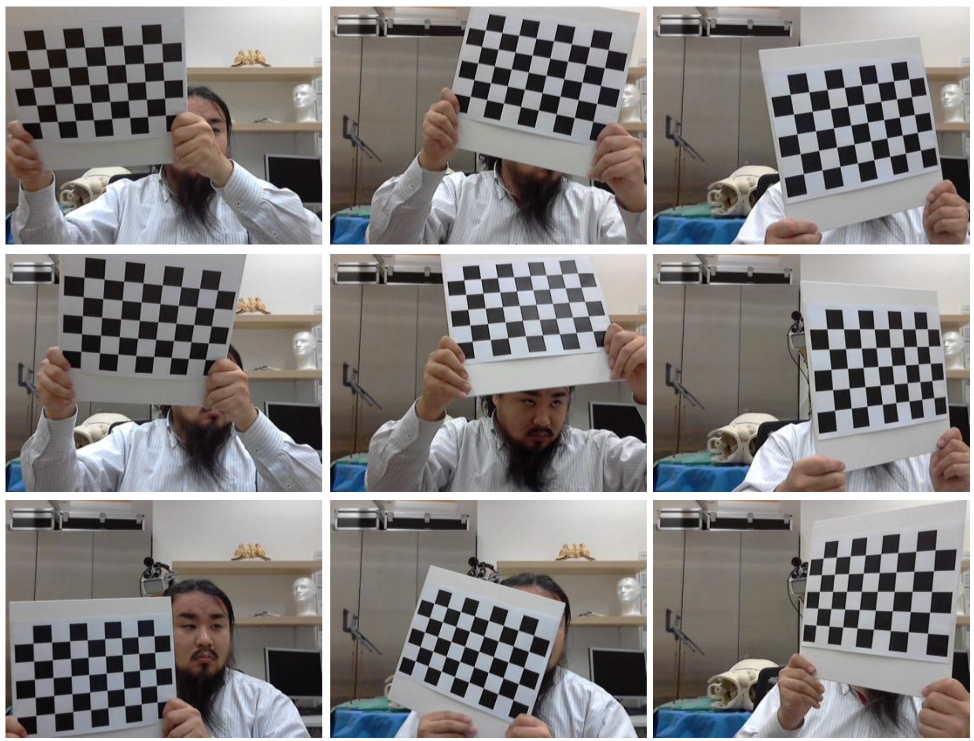


MatLab Camera Calibration Toolbox [3]

[26] Z. Zhang, ICCV, 1999
[3] J. Y. Bouget, MatLab Camera Calibration Toolbox, 2008

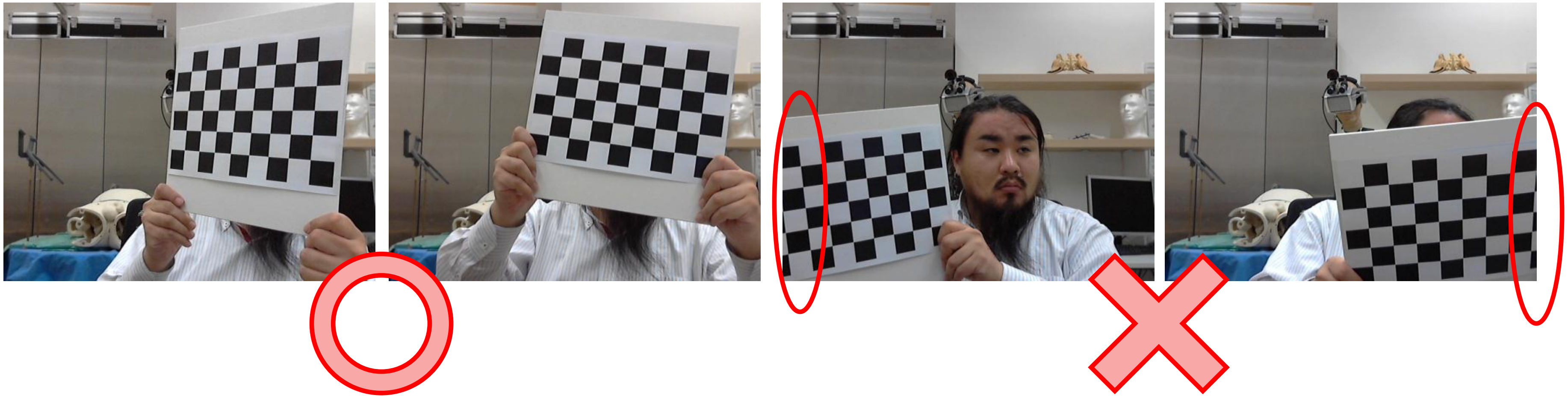
For accurate calibration…

- Calibration object should be 3d:
  - Fill entire view volume
  - Different poses >> same pose
  - Different depths >> same pose



Our dilemma

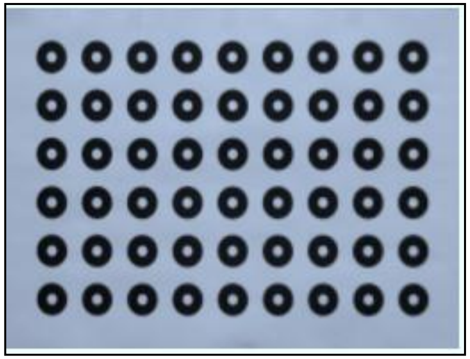
- Strong assumption: entire object must be visible.

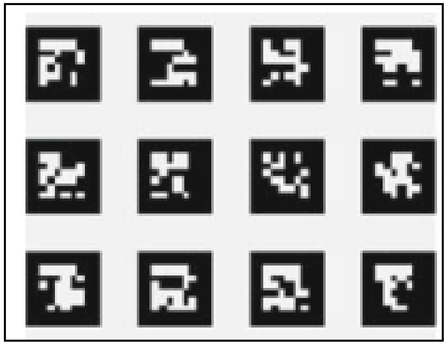


- For localization: hesitates to go closer to image border.
- For accuracy: better to go as close to image border as possible.

Literature: points correspondence

	Circle/rings grid [4]	AR Tag [7]	Natural image [19]
Occlusion	X	O	O
Defocus	O	X	O
Perspective distortion	△	O	△



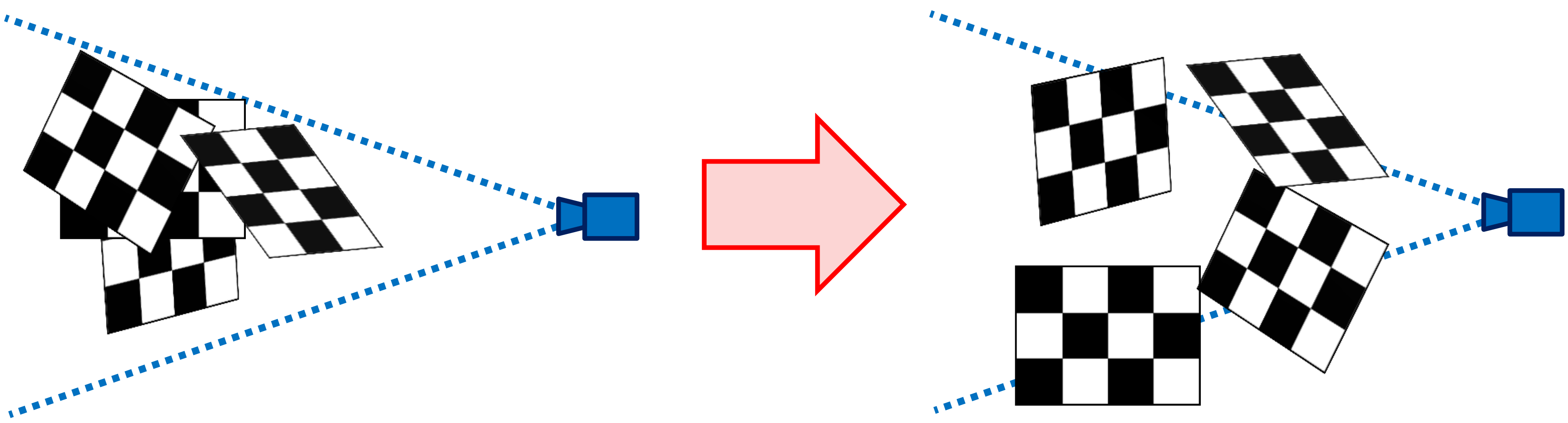


[4] Datta, ICCV workshop, 2009
[7] Fiala, MVA, 2008
[19] Pilet, ISMAR, 2006

Motivation

- Remove the assumption = Handle partial occlusion.

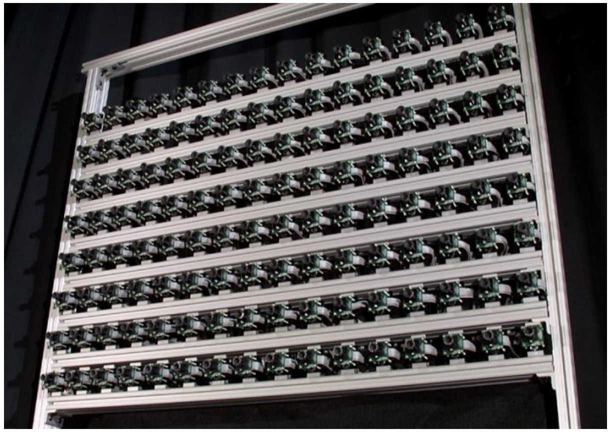
- More accurate estimation by filling view volume.
- Less frustration during image acquisition.



Motivation

- Especially, useful for multiple cameras calibration (though this work focuses on single camera calibration...)

Distributed cameras
Eyevision, CMU



Hundreds of cameras
The Stanford Multi-Camera Array

Different types of cameras
HMD based AR, TUM

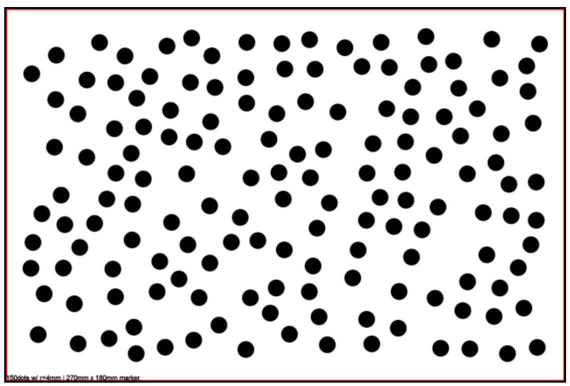
The proposed method

Our idea

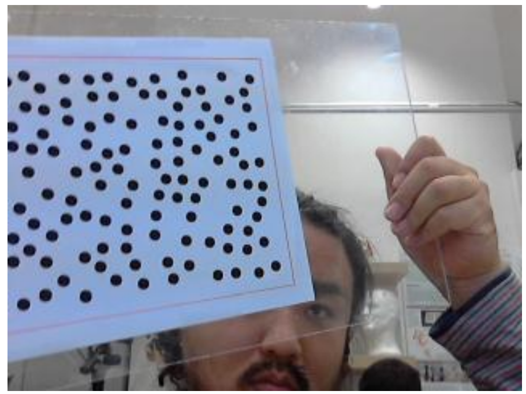
- Idea: Uses state-of-the-art marker tracking algorithm.
  - Automatic detection and localization even with partial occlusion. = Can put calibration object closer to image border.
- so that
  - More accurate estimation on lens distortion parameters.
  - Flexible calibrations for vision based systems.

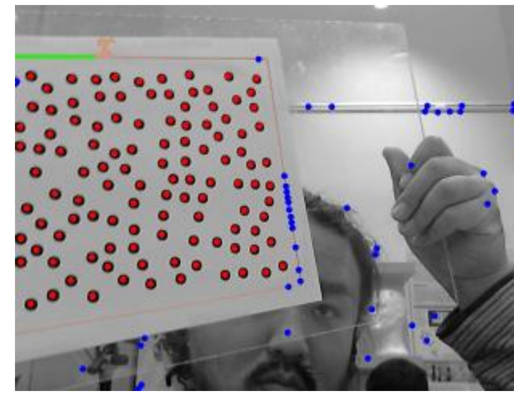
The proposed method

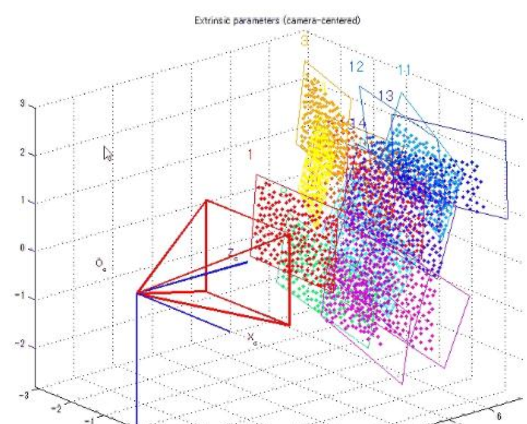
- Points correspondence: applies tracking algorithm on the images.
- Parameter estimation: optimizes using the points correspondence.



RANdom DOts Marker







Method 1/2: overview

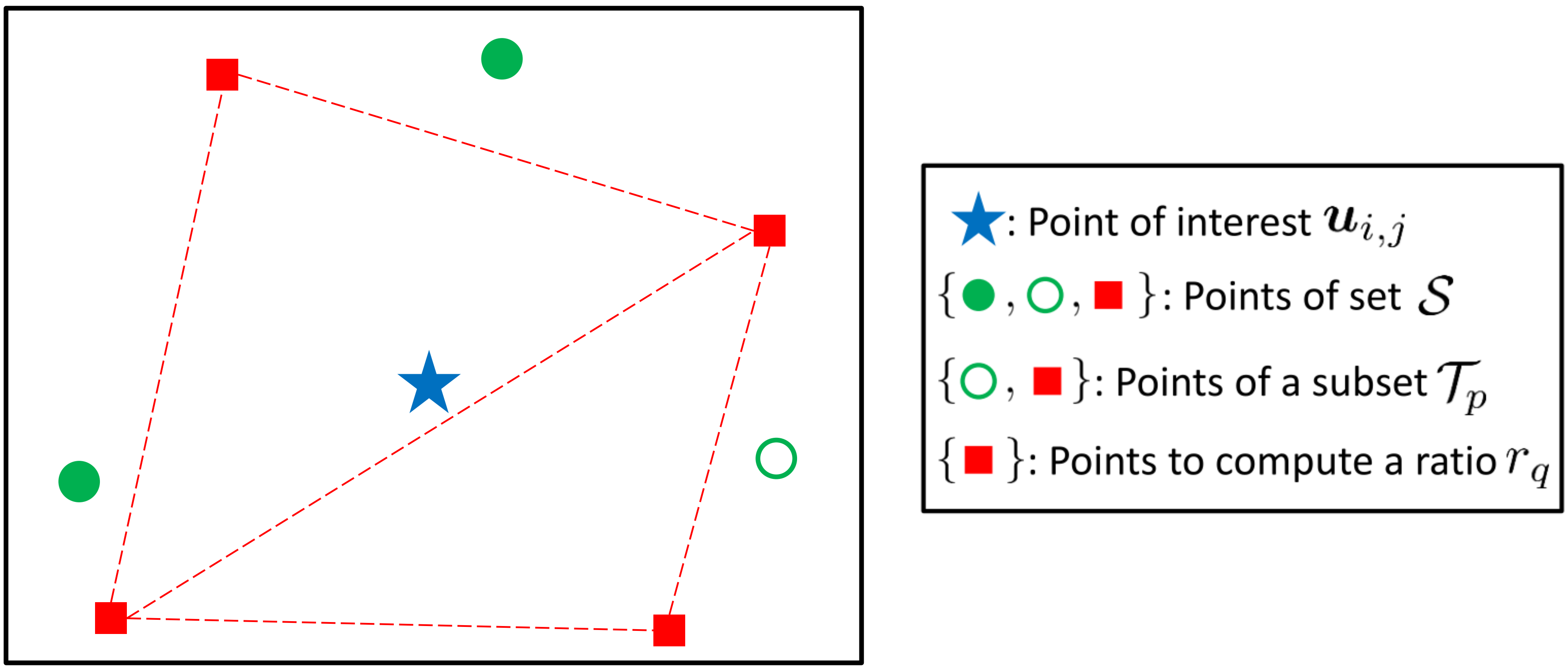
- Applies RANDOM tracking algorithm [22] on the images.
  - Simple circle detection.
  - Fast matching using LLAH.

Input | Circle detection | Points matching

[22] Uchiyama, IEEE VR, 2011

Method 1/2: tracking algorithm

- Tracking algorithm [22] invariant to scale & rotation.
  - Use distribution of control points.
  - For a point of interest, set of ratio of two triangles consists of its neighboring points.



[22] Uchiyama, IEEE VR, 2011

Method 2/2: parameter estimation

- Based on Zhang's method [26].
  - Non-linear optimization on reprojection error of control points.
  - Consider the visibility of control points.

$$\sum_{i \in \mathcal{I}_{\text{homo}}} \sum_{j=1}^{M} \left\| \tilde{\boldsymbol{x}}_{i,j} - \text{Proj}(\tilde{\boldsymbol{X}}_j, \boldsymbol{A}, \boldsymbol{\delta}, \boldsymbol{R}_i, \boldsymbol{t}_i) \right\|^2 .$$

$$\sum_{i \in \mathcal{I}_{\text{homo}}} \sum_{j=1}^{M} v_{i,j} \left\| \tilde{\boldsymbol{x}}_{i,j} - \text{Proj}(\tilde{\boldsymbol{X}}_j, \boldsymbol{A}, \boldsymbol{\delta}, \boldsymbol{R}_i, \boldsymbol{t}_i) \right\|^2 . \quad (11)$$

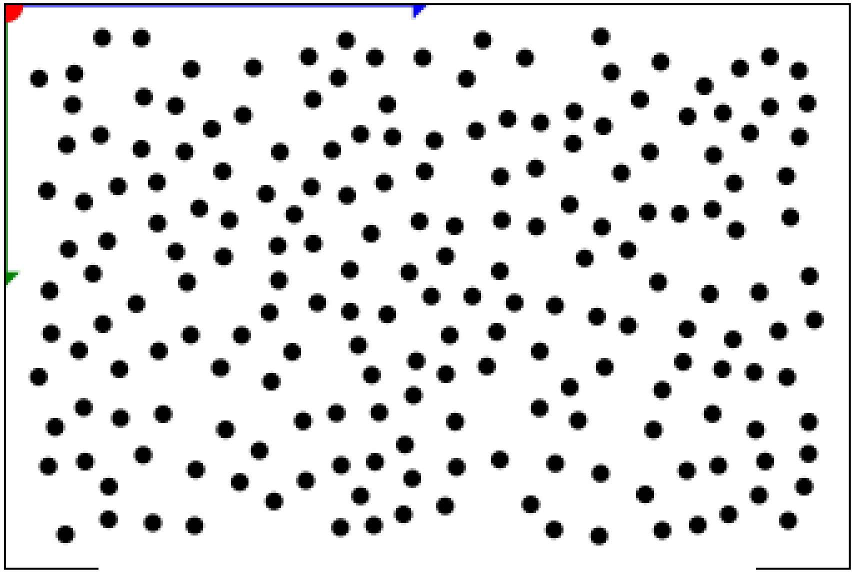
$$v_{i,j} = \begin{cases} 1 & \text{if the point is visible} \\ 0 & \text{otherwise} \end{cases}$$

[26] Z. Zhang, ICCV, 1999

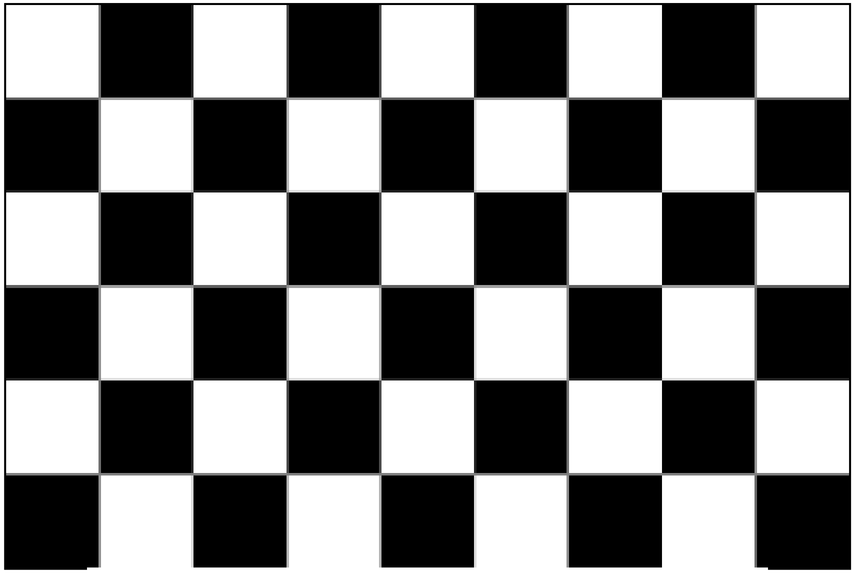
Experimental results

Experiments

- Simulation experiments:
  - RANDOM with 200 control points.
  - Chessboard with 40 control points
- Real world experiment:
  - RANDOM with 200 control points.



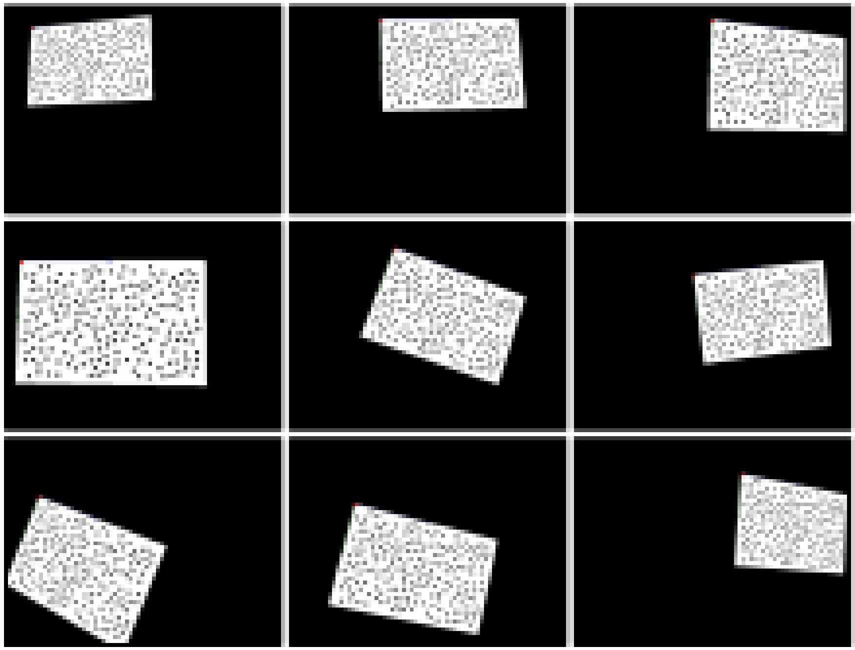
200 points on RANDOM



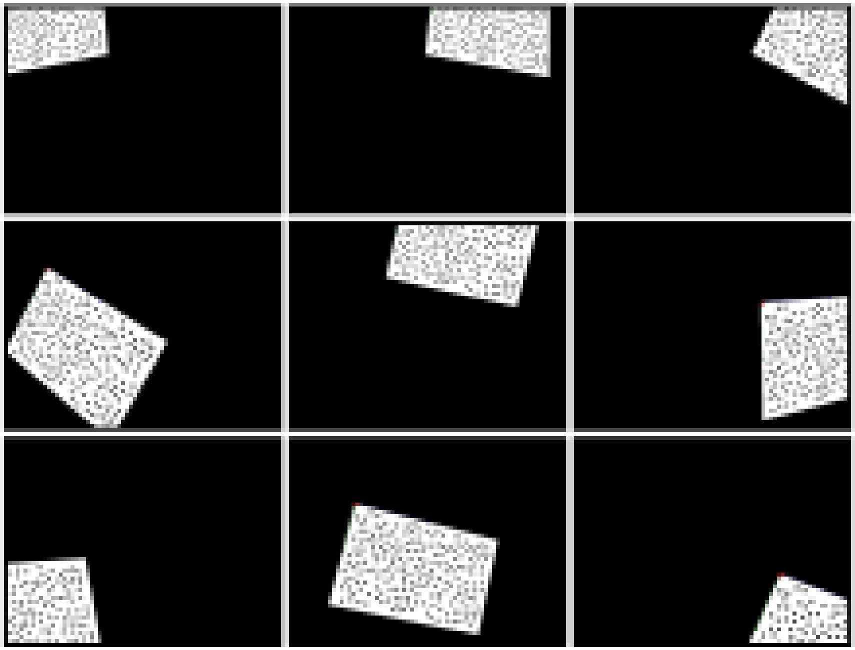
40 points on chessboard

Simulation experiment 1/2

- Q: More accurate result with marker located around image border?
- Comparison:
  1. 10 images from (a).
  2. 10 images from (a) + 1-10 images from (b).



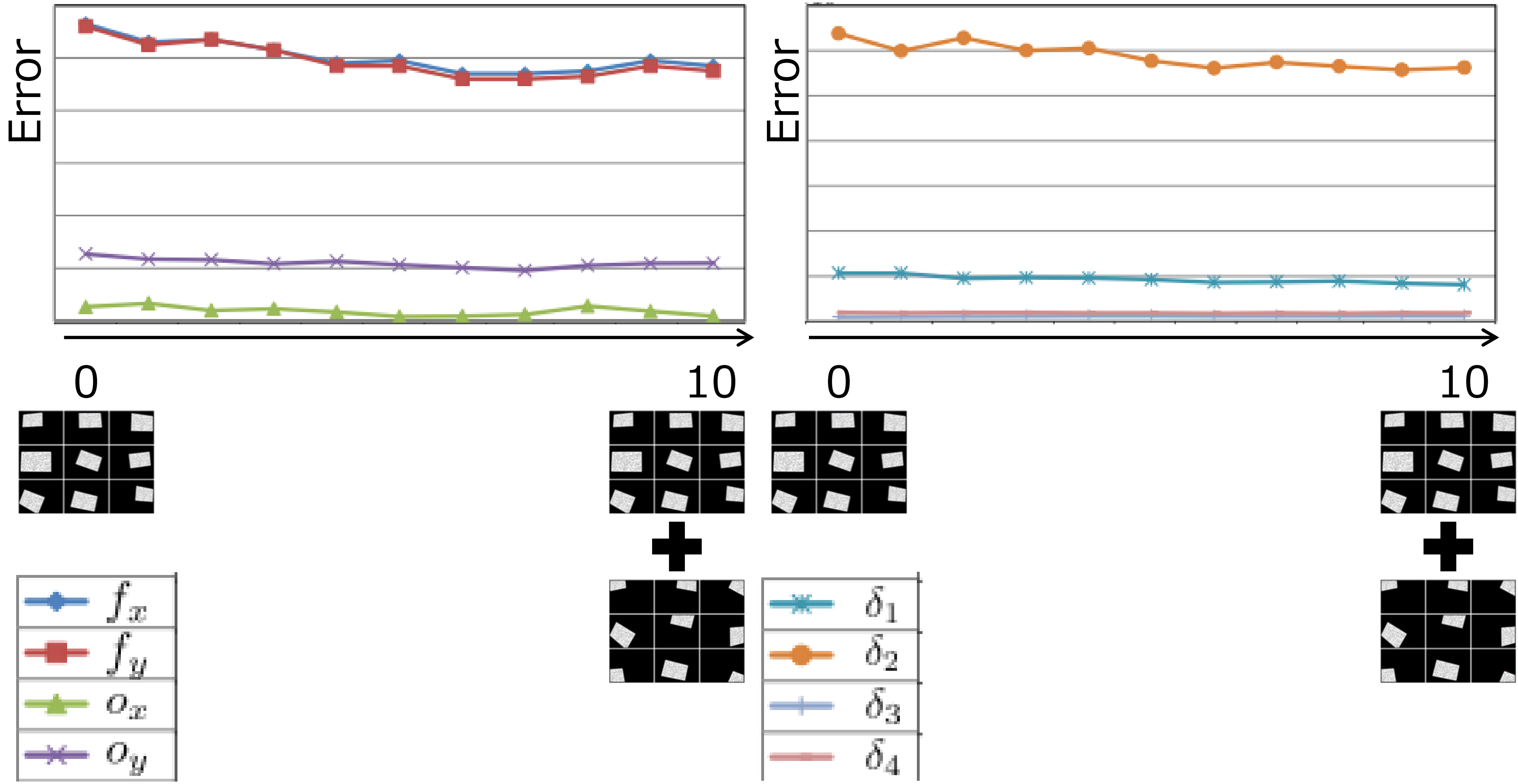
(a) Entire RANDOM



(b) Partial RANDOM

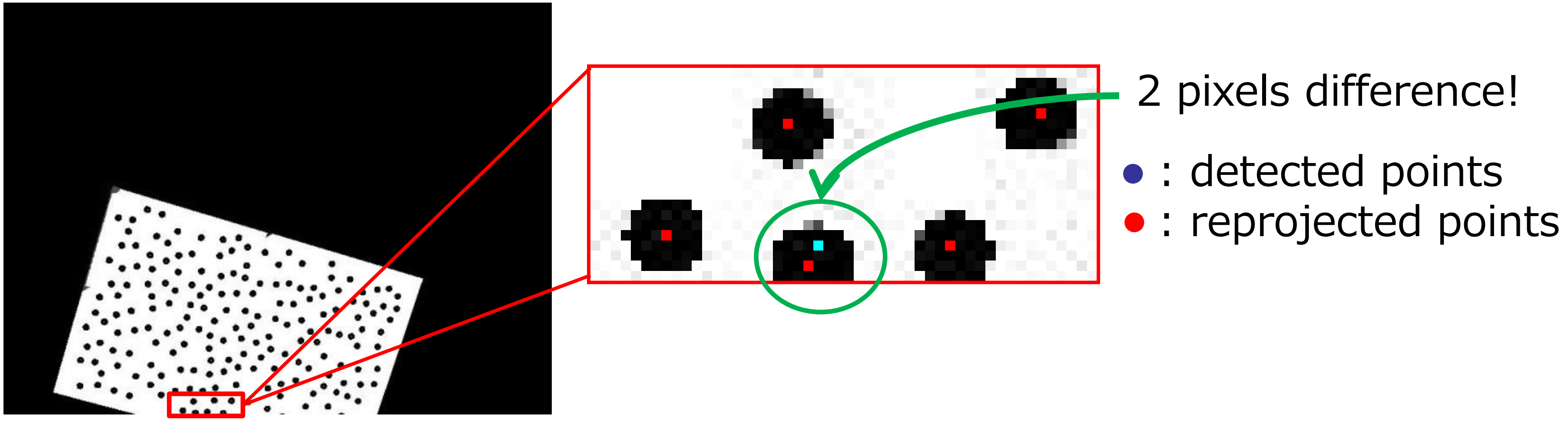
Simulation experiment 1/2: result

- Error: mean value



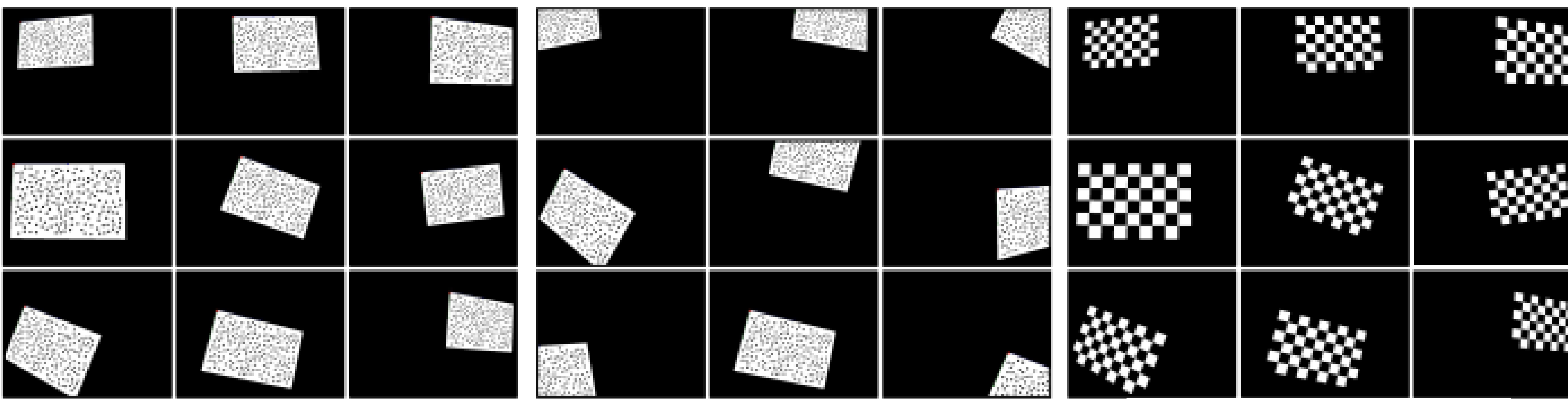
Simulation experiment 1/2: discussion

- Two disadvantages:
  - Center of circle is not perspective invariant.
  - Inaccurate detection due to poor circle detection algorithm.



Simulation experiment 2/2

- Q: Is the proposed method better than chessboard one?
- Comparison:
  1. 10 images from (a).
  2. 10 images from (a) + 10 images from (b).
  3. 10 images from (c).
  4. 20 images from (c).



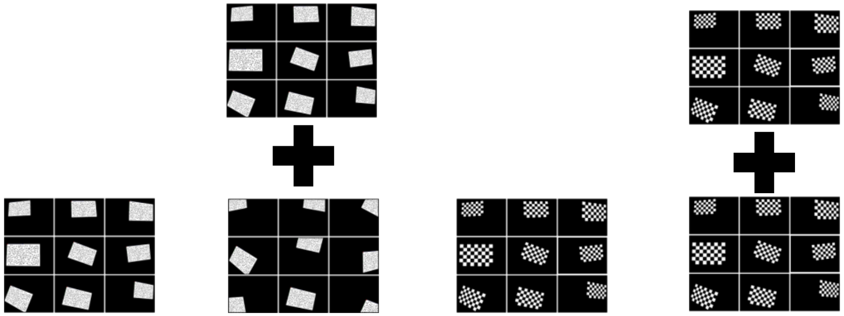
(a) Entirely visible RANDOM

(b) Partially visible RANOM

(c) Entirely sible chessboard

Simulation experiment 2/2: result

- Our method results less reprojection error.



		Ours		Chessboard	
Test case		(1)	(2)	(3)	(4)
x axis	Mean [$\times 10^{-1}$]	**0.71**	0.72	1.67	1.70
	Std. dev. [$\times 10^{-2}$]	0.32	**0.25**	2.00	1.14
y axis	Mean [$\times 10^{-1}$]	**0.70**	0.72	1.65	1.67
	Std. dev. [$\times 10^{-2}$]	0.34	**0.24**	2.24	1.24

Simulation experiment 2/2: result

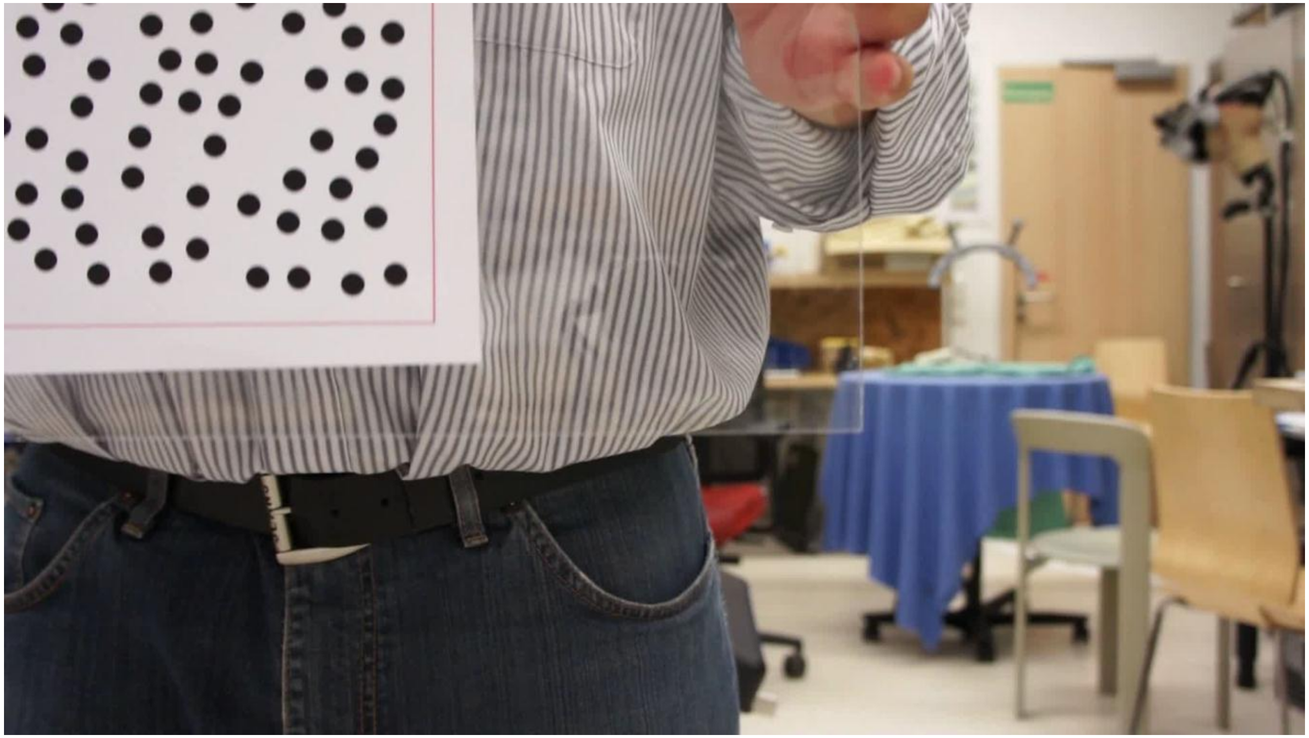
- It's strange that case 1 outperforms case 3 because
  - control points on a chessboard is perspective invariant,
  - control points on RANDOM is
    - scale & rotation invariant,
    - center of circle is not perspective invariant.

- The result may be due to number of control points.
  - 200 control points on RANDOM
  - 40 control points on chessboard

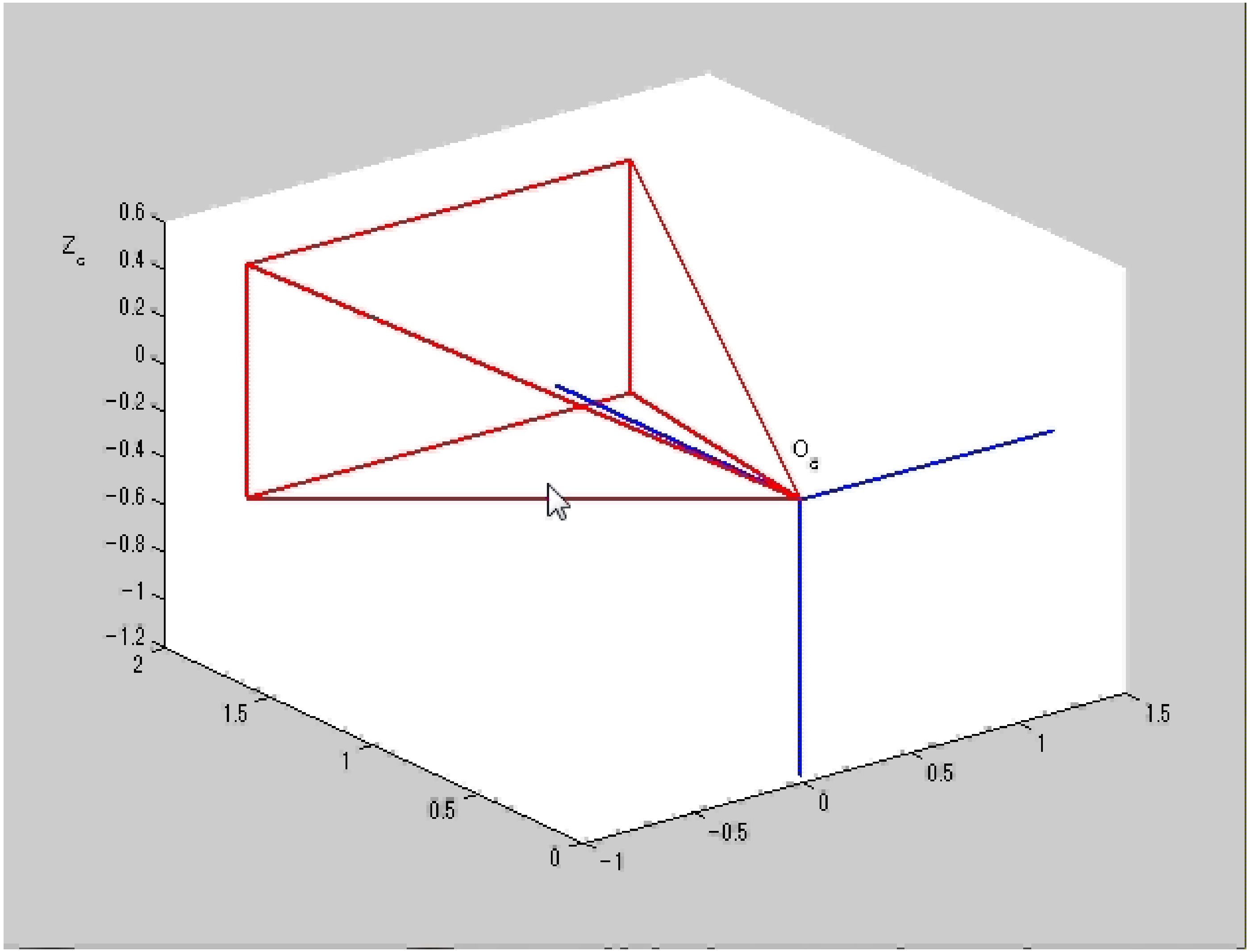
- Will perform more fair comparison...

Real world experiments 1/2

- Single camera calibration: Sony Nex-5.
  - Resolution: 1920x1080
  - Number of images: 100
  - Number of control points on RANDOM: 200

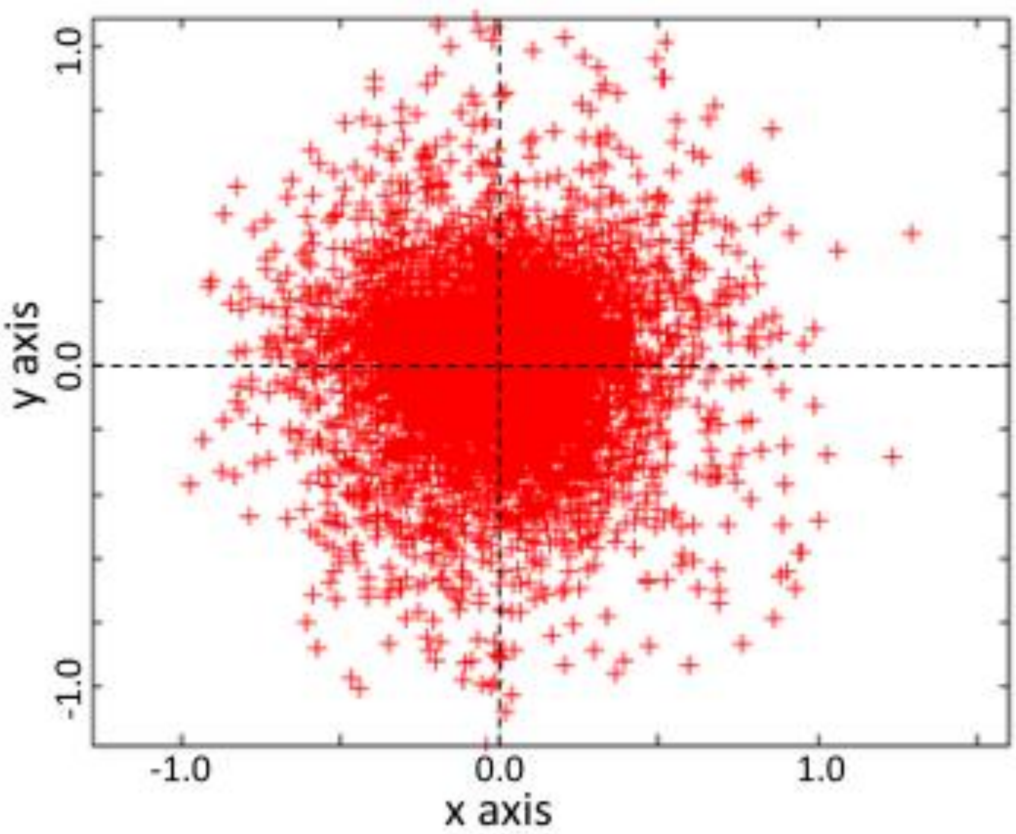


Real world experiments 1/2: result



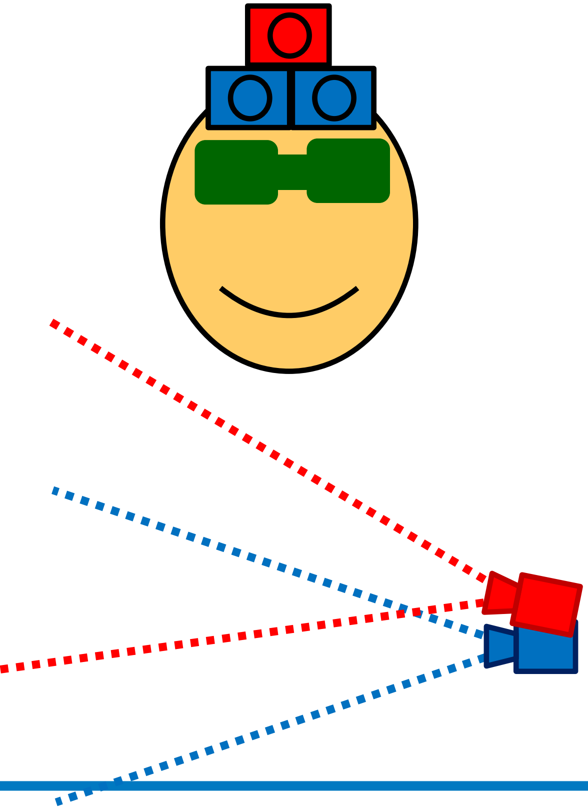
Real world experiments 1/2: result

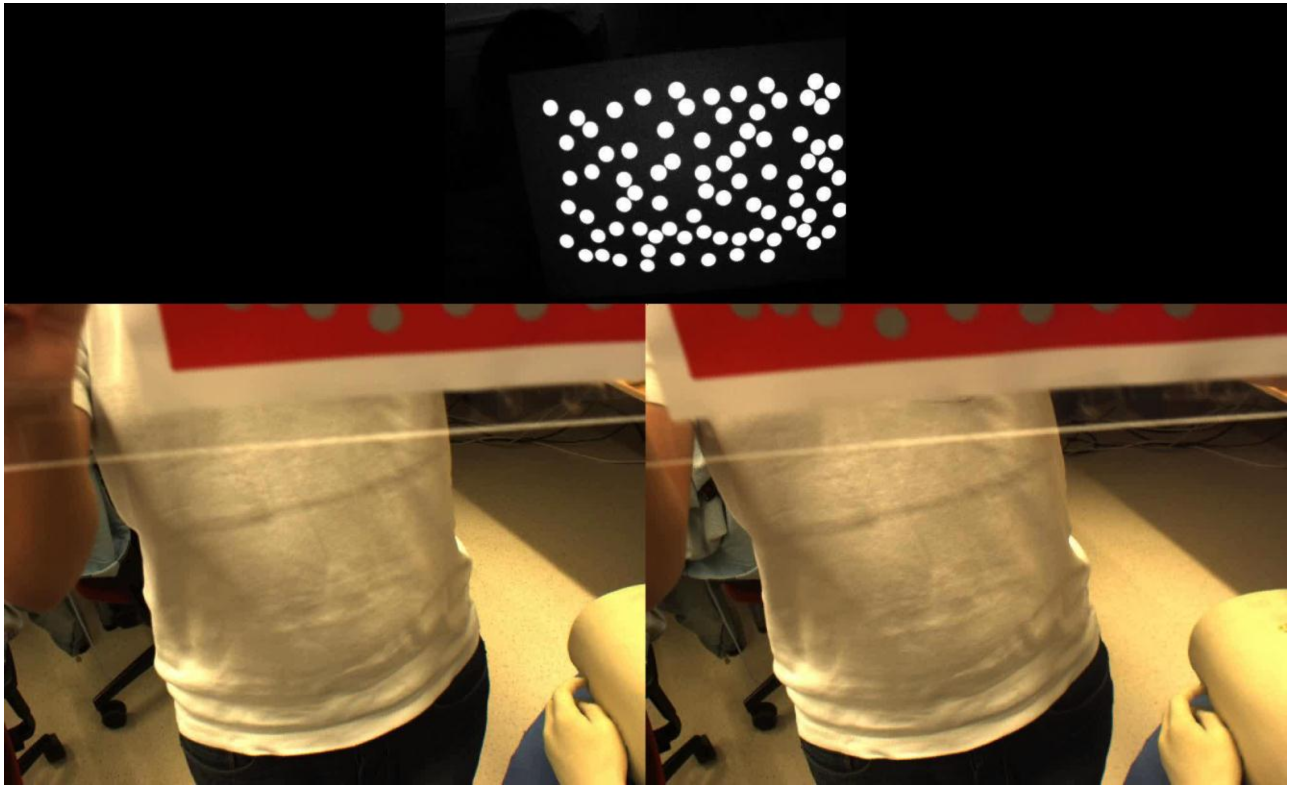
- Reprojection error: 0.16 ± 0.15 pixel
- Maximum error: 1.30 pixels



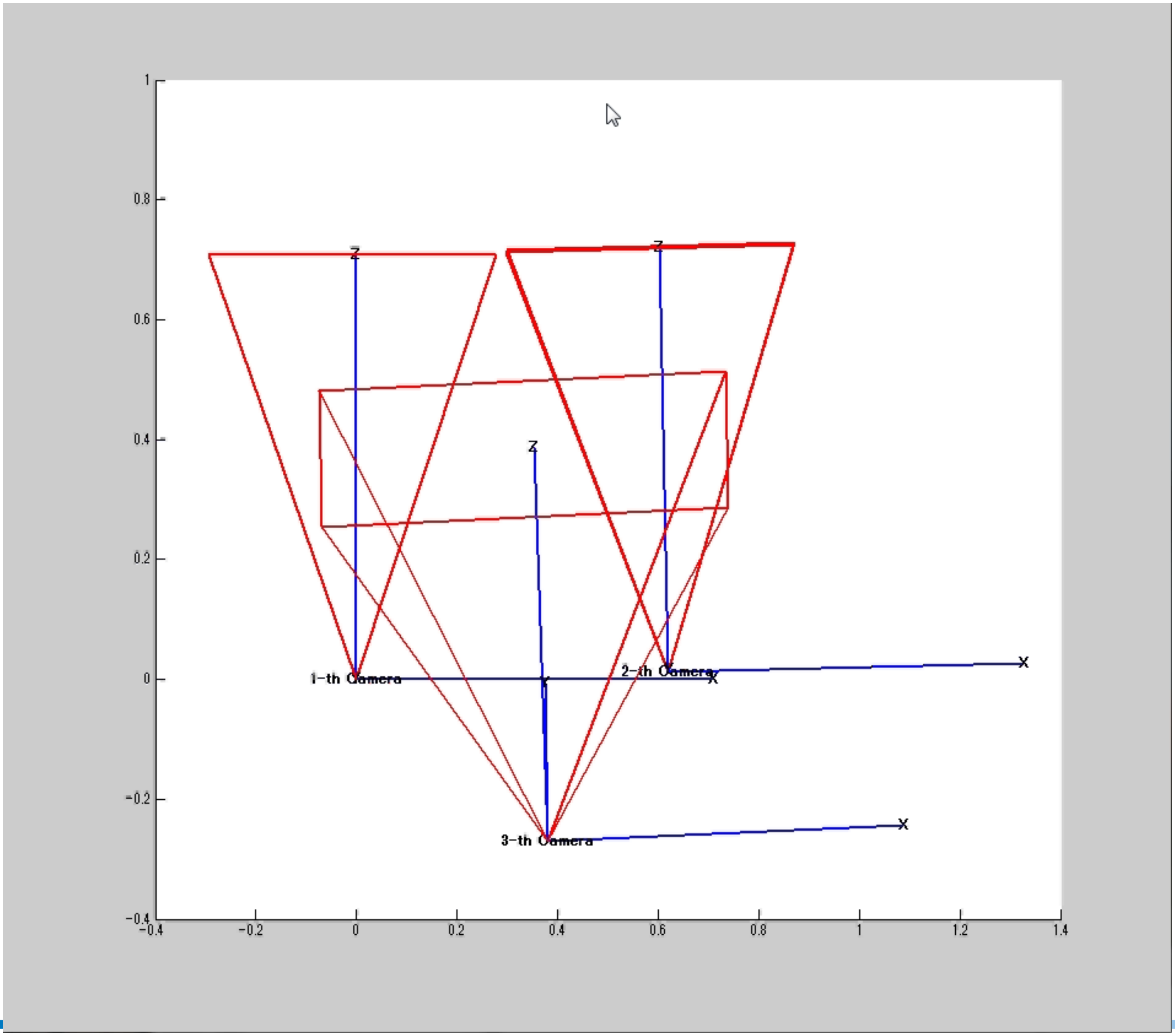
Real world experiments 2/2

- Multiple cameras calibration (3 cameras attached on an HMD):
  - 1 IR camera
  - 2 color cameras





Real world experiments 2/2: result



Conclusion
+
future works

Conclusion

- Used marker tracking algorithm
- To solve points correspondence problem
- For more accurate & friendly camera calibration.

- Advantage:
  - More accurate & stable calibration result.
  - Many potential extensions.

- Limitation:
  - The tracking algorithm is only scale & rotation invariant.
    = heavy rotation along x/y axis is not supported.
  - Center of circle is not perspective invariant.

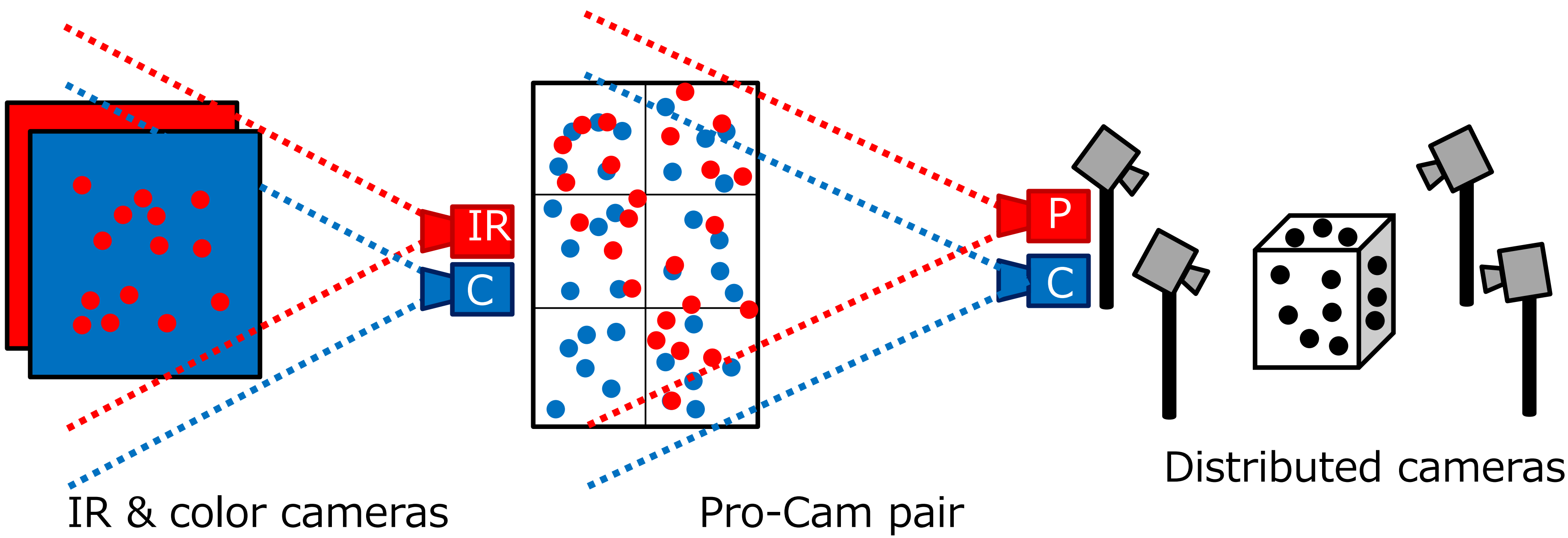
Potential extension

- Multiple cameras calibration.
- Multiple markers for calibration [A1].

[A1] GML Camera Calibration Toolbox, 2005

Potential applications

- Different types of cameras: combination of color & IR markers
- Projector-camera: one printed and one projected markers.
- Distributed cameras: multiple markers.



Future works: tracking for points correspondence

- Use perspective invariant metric for points correspondence.
  - line segments tracking [A2].

- Center of circle is not perspective invariant.
  - Use perspective invariant mark [A3].

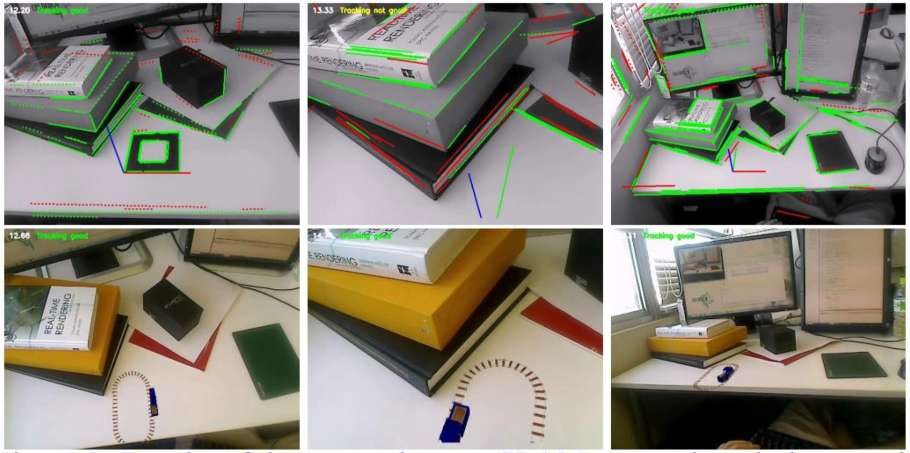
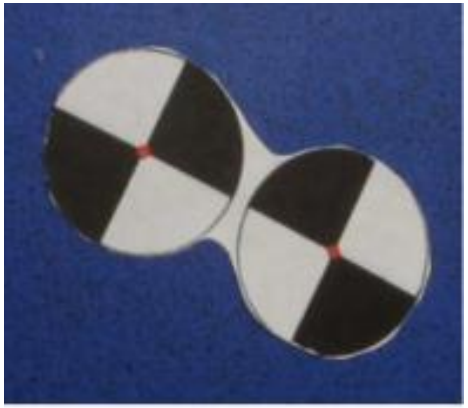


Figure 2: Results of demonstrating our SLAM system in a desktop environment.

Line segments tracking_ for SLAM [A2]

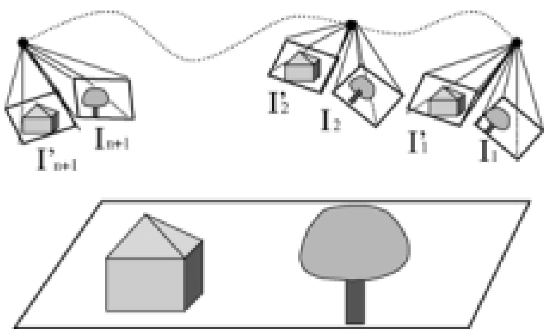


Perspective invariant mark [A3]

[A2] Hirose, BMVC, 2012
[A3] Beeler, Tech. rep. ETHZ, 2010

Future works: tracking for calibration constraints

- Uses rigidity of cameras for non-overlapping cameras calibration.
  - Tracking for knowing each camera motion,
  - Then align the unsynchronized cameras using their rigidity [A4].



- Selects good calibration images from long video sequences.
  - Somehow evaluate calibration images
  - To reduce unnecessary huge amount of images from video sequences.

[A4] Irani, IJCV, 2002

Code available

- Entire package containing tracking and calibration by me.
  - will publish as an open source
  - current version: C++ + MatLab
  - future version: C++
  - If you want to use it, please contact me!

- Original RANDOM tracking algorithm by Hideaki Uchiyama [22].
  - open source
  - C++

- User friendly calibration code on github by Alexandru Duliu.
  - open source
  - C++

Thanks for your attention...

- I'm looking for a job opportunity.
- present-03.2013: postdoc @ Keio Univ. & visiting postdoc @ TUM
- 03.2012-??: not decided yet...

- My research interests
  - camera tracking for practical application,
  - image restoration
    - deblurring
    - focus control
  - 3d modeling
    - 3d reconstruction using depth camera
    - photometric stereo
  - AR visualization to improve perception

Acknowledgments

- This work was partially supported by the Strategic Young Researcher Overseas Visits Program for Accelerating Brain Circulation of Japan Society for the Promotion of Science, G2308