# dVRK stereo camera calibration and model registration

## *-Seminar Presentation-*

Presented by: Peter Ahn

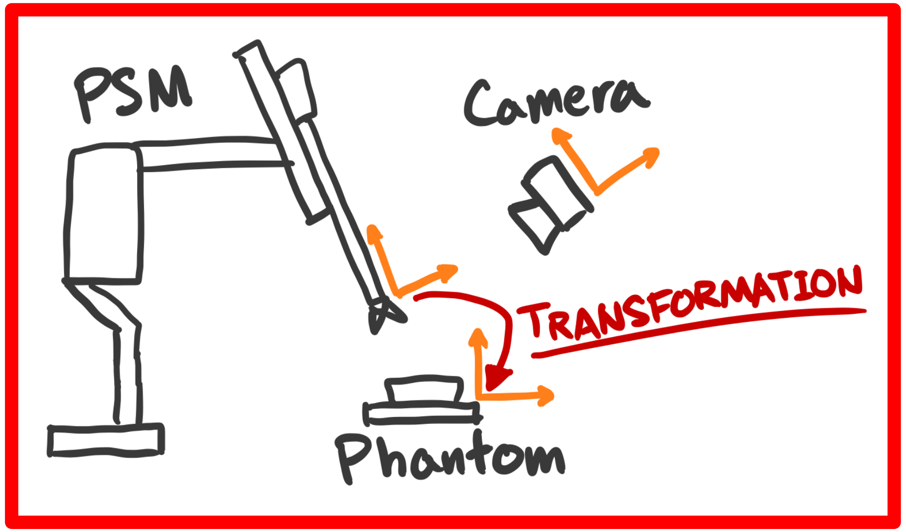
Group 12: Peter Ahn and Mengze Xu
Mentors: Preetham Chalasani and Anton Deguet
May 2, 2017

# Quick Review

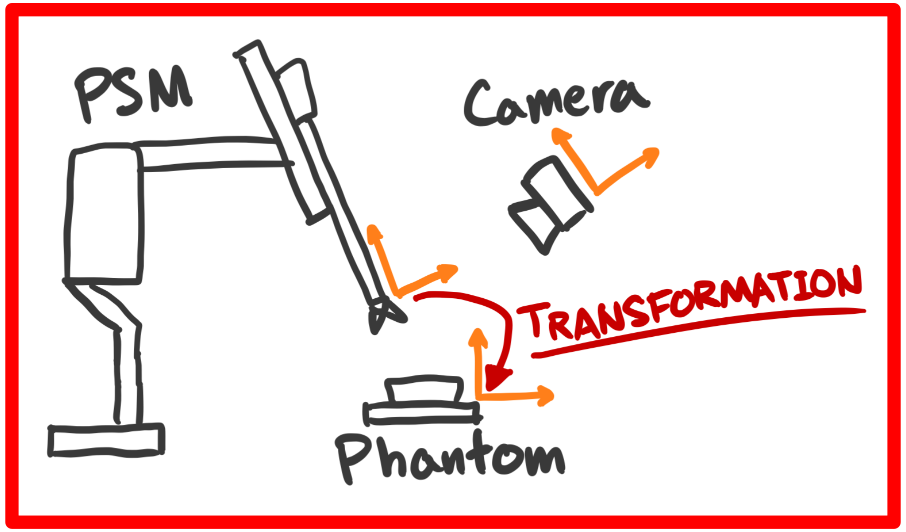
- Goal: Register surfaces to the robot (Patient Side Manipulator, PSM)

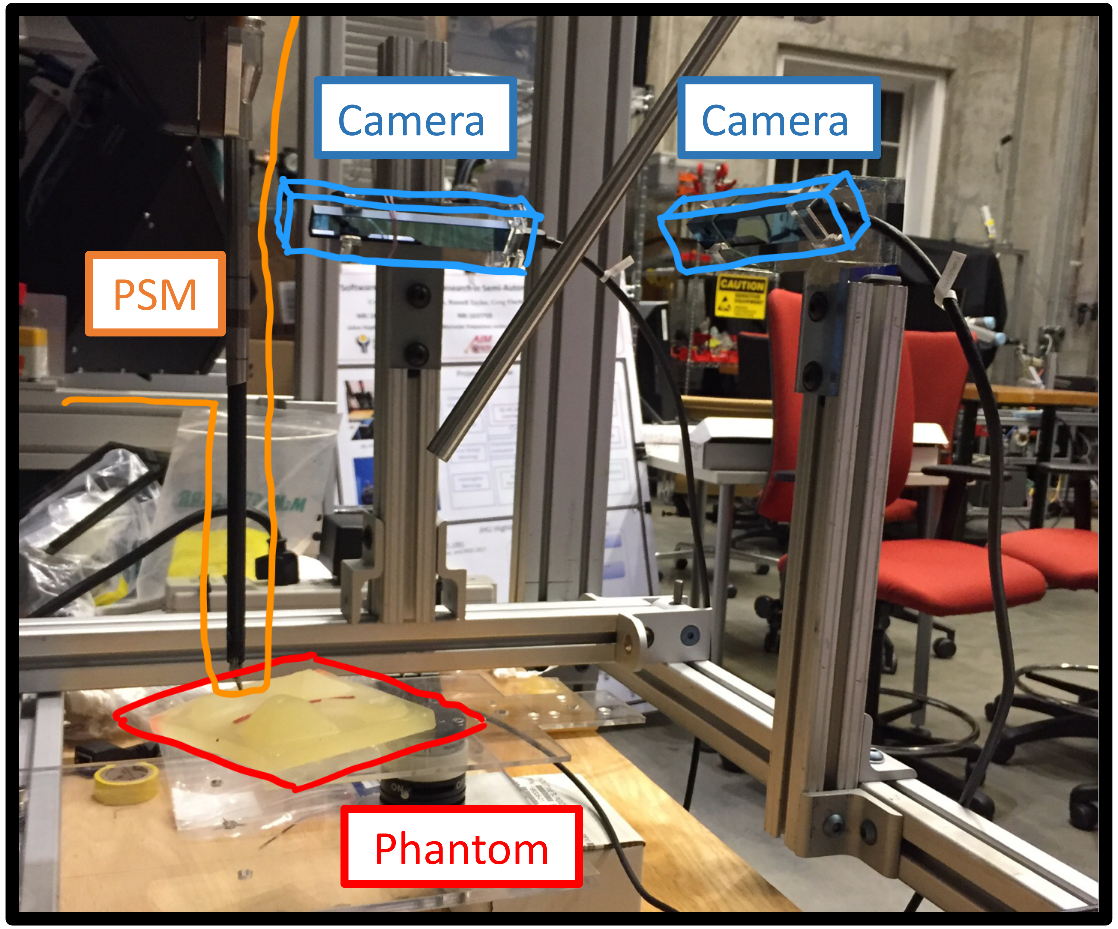
- Current Method: Move robot to touch the surface

- What we want to do: Using calibrated stereo camera to substitute touching



# Quick Review



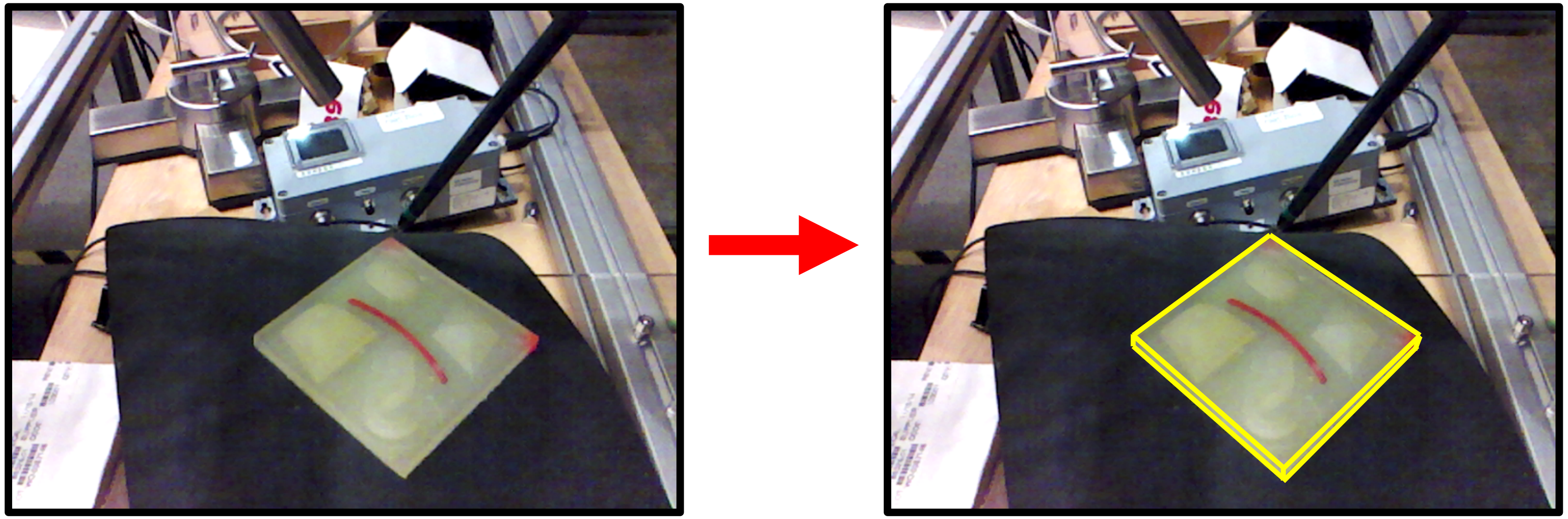

- Complete Hand-Eye Calibration
- Detect and register phantom surface to PSM
- Desired error under 1~2 *millimeters*

# Paper Selection

[1] Azad P., Asfour T., Dillmann R. “Stereo-Based vs. Monocular 6-DoF Pose Estimation Using Point Features: A Quantitative Comparison.” In: *Dillmann R., Beyerer J., Stiller C., Zöllner J.M., Gindele T. (eds) Autonome Mobile Systeme 2009.* Informatik aktuell. Springer, Berlin, Heidelberg

**Stereo-Based vs. Monocular 6-DoF Pose Estimation Using Point Features: A Quantitative Comparison**

# Relevance to our project



- We need accurate pose estimation of the phantom in 3D space
  → For our expected deliverable (Known Surface Registration)

- In paper: quantitative comparison between two different methods
  → **Monocular vs. Stereo Camera System**

# Paper Background

Stereo-Based vs. Monocular 6-DoF Pose Estimation Using Point Features: A Quantitative Comparison

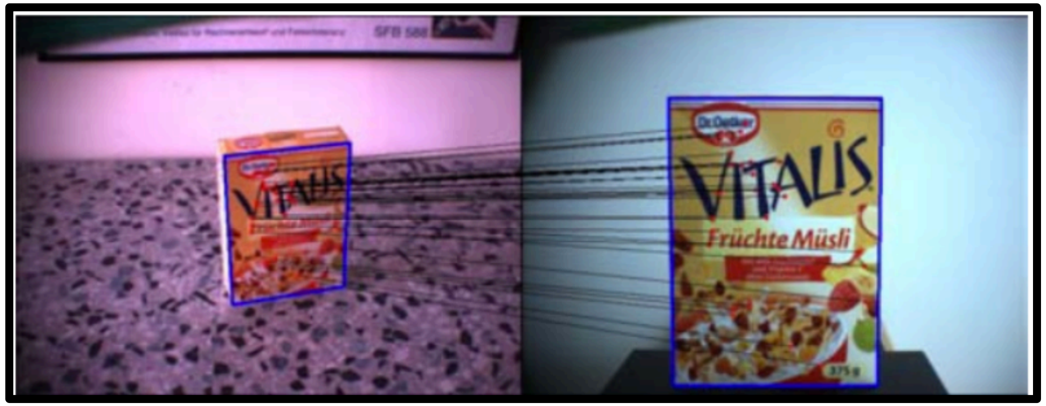
- Importance of accurate pose estimation of objects in 3D space, especially for robotic manipulation applications

- Limits of monocular approach that uses 2D-3D correspondences

- Two different approaches to computing a 6-DoF pose:
  - monocular vs. stereo-based pose estimations

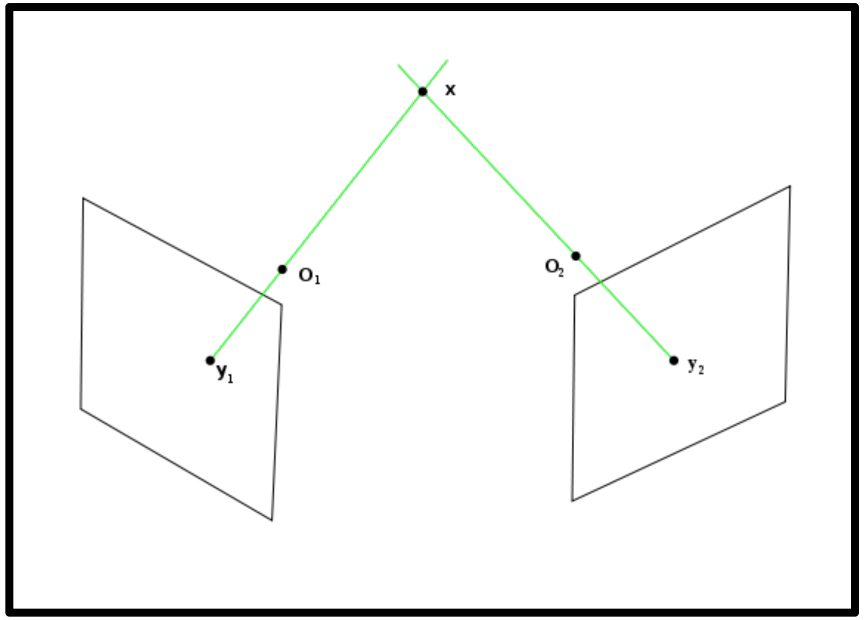
# Theoretical Accuracy Comparison

Monocular Pose Estimation
→ 2D-3D point correspondences




[4]

Stereo-based Pose Estimation
→ 3D calculations using stereo triangulation




https://en.wikipedia.org/wiki/Triangulation_(computer_vision)
#/media/File:TriangulationIdeal.svg

# Theoretical Accuracy Comparison

Focal length = 4 mm → $f = 530\ pixels$
Baseline $b$ = 90 mm (principal axes parallel)
Largest distance between feature pair = 100 mm

**Monocular Pose Estimation**

$$\frac{z_c(u)}{z_c(u+\ \Delta)} - 1 = \frac{\Delta}{u}$$ [1]

$$u = \frac{f * x_c}{z_c} \approx 70; Pixel\ Error\ \Delta = 1$$

$Projected\ size\ u\ increase, error\ decrease$

$$75\ cm * \frac{1}{70} \approx \mathbf{1\ cm}$$

**Stereo-based Pose Estimation**

$$\frac{z_c(d)}{z_c(d+\ \Delta)} - 1 = \frac{\Delta}{d}$$ [1]

$$d = \frac{f * b}{z_c} \approx 64; Pixel\ Error\ \Delta = 0.5$$

$Disparity\ d\ increase, error\ decrease$

$$75\ cm * \frac{0.5}{64} \approx \mathbf{0.6\ cm}$$

# 6-DoF Pose Estimation

Monocular Pose Estimation
→ 2D-3D point correspondences

Stereo-based Pose Estimation
→ 3D calculations using stereo triangulation

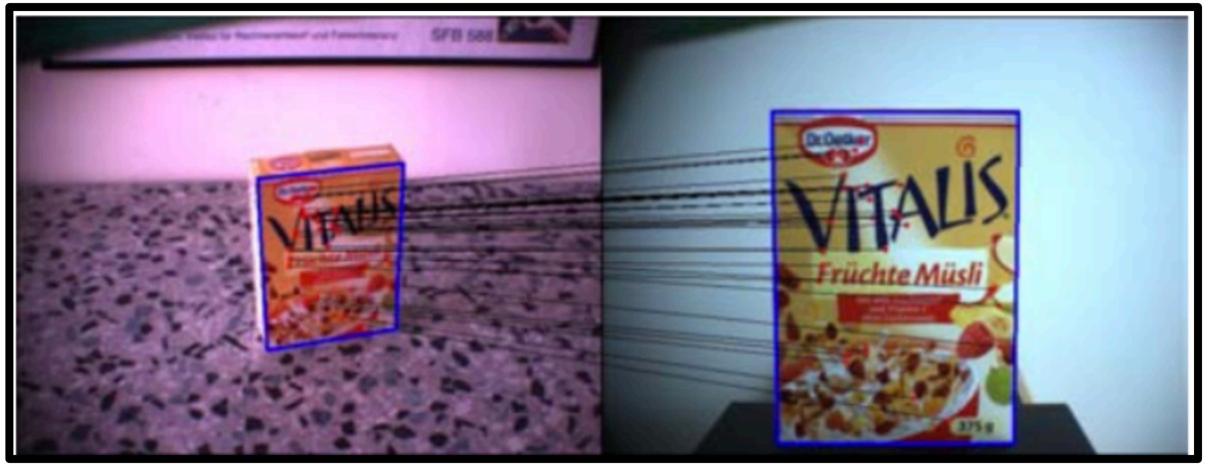
**Algorithm 1** CalculatePoseTextured($I_l$, $I_r$, $C$) → $R, \boldsymbol{t}$

1. Determine the set of interest points within the calculated 2D contour $C$ of the object in the left camera image $I_l$.
2. For each calculated point, determine a correspondence in the right camera image $I_r$ by computing the *Zero Normalized Cross Correlation* (ZNCC) along the epipolar line.
3. Calculate a 3D point for each correspondence.
4. Fit a 3D model of the object into the calculated 3D point cloud and return the resulting rotation $R$ and the translation $\boldsymbol{t}$.

# 6-DoF Pose Estimation

**Algorithm 1** CalculatePoseTextured($I_l$, $I_r$, $C$) $\rightarrow R, \boldsymbol{t}$

1. Determine the set of interest points within the calculated 2D contour $C$ of the object in the left camera image $I_l$.
2. For each calculated point, determine a correspondence in the right camera image $I_r$ by computing the *Zero Normalized Cross Correlation* (ZNCC) along the epipolar line.
3. Calculate a 3D point for each correspondence.
4. Fit a 3D model of the object into the calculated 3D point cloud and return the resulting rotation $R$ and the translation $\boldsymbol{t}$.

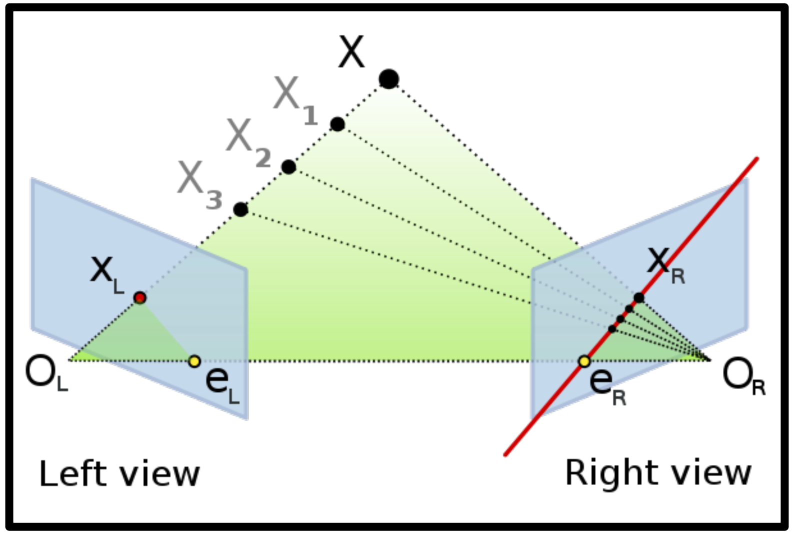



[4]

# 6-DoF Pose Estimation

**Algorithm 1** CalculatePoseTextured($I_l$, $I_r$, $C$) $\rightarrow R, \boldsymbol{t}$

1. Determine the set of interest points within the calculated 2D contour $C$ of the object in the left camera image $I_l$.
2. For each calculated point, determine a correspondence in the right camera image $I_r$ by computing the *Zero Normalized Cross Correlation* (ZNCC) along the epipolar line.
3. Calculate a 3D point for each correspondence.
4. Fit a 3D model of the object into the calculated 3D point cloud and return the resulting rotation $R$ and the translation $\boldsymbol{t}$.

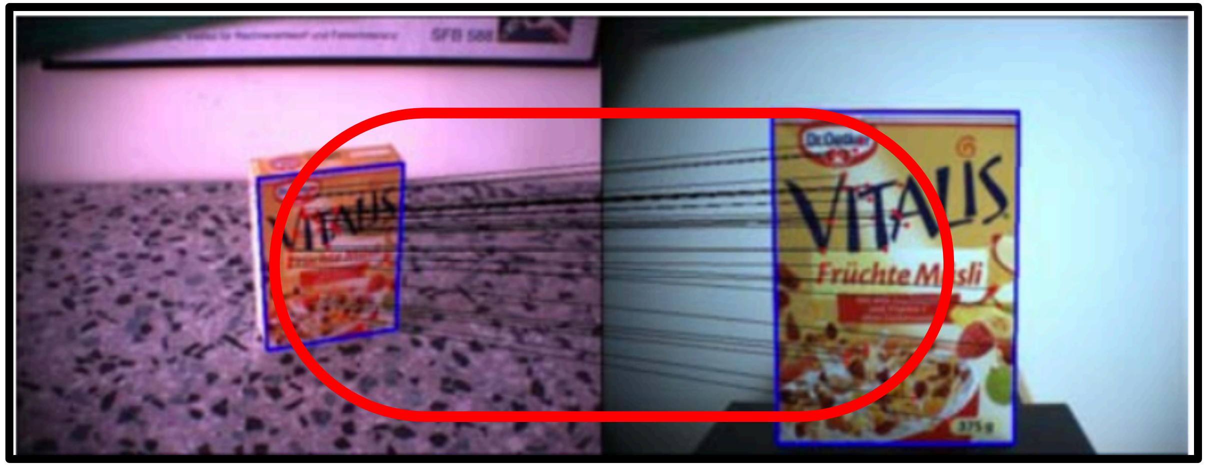



https://en.wikipedia.org/wiki/Epipolar_geometry#/media/File:Epipolar_geometry.svg

# 6-DoF Pose Estimation

**Algorithm 1** CalculatePoseTextured($I_l$, $I_r$, $C$) $\rightarrow R, \boldsymbol{t}$

1. Determine the set of interest points within the calculated 2D contour $C$ of the object in the left camera image $I_l$.
2. For each calculated point, determine a correspondence in the right camera image $I_r$ by computing the *Zero Normalized Cross Correlation* (ZNCC) along the epipolar line.
3. Calculate a 3D point for each correspondence.
4. Fit a 3D model of the object into the calculated 3D point cloud and return the resulting rotation $R$ and the translation $\boldsymbol{t}$.

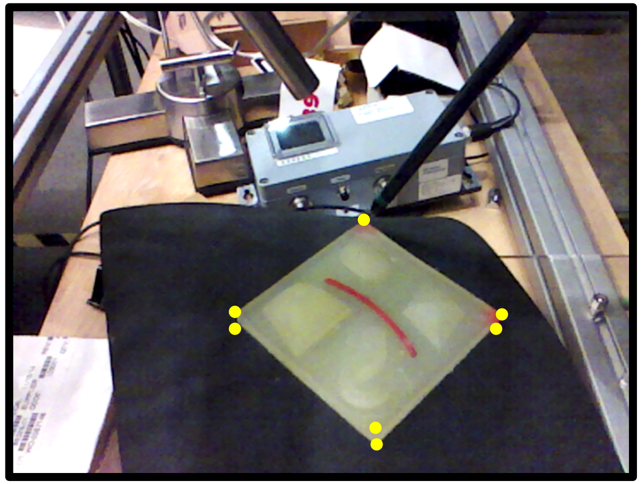


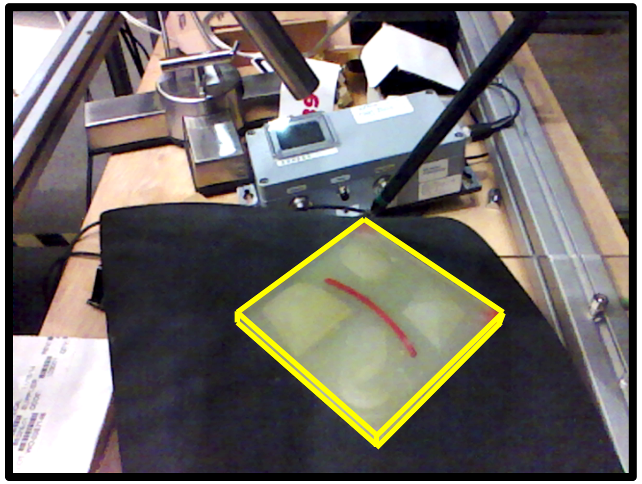

[4]

# 6-DoF Pose Estimation

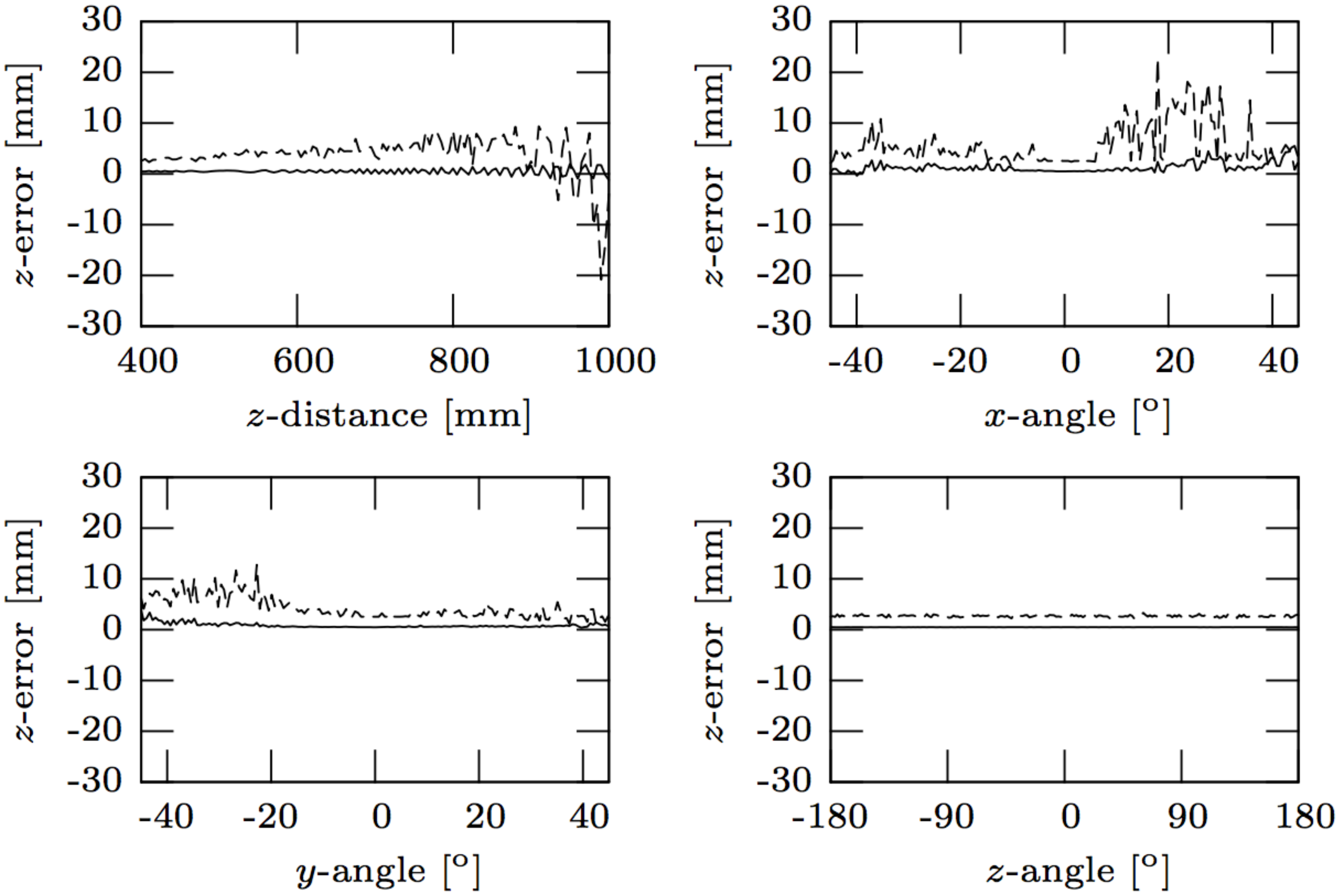
**Algorithm 1** CalculatePoseTextured($I_l$, $I_r$, $C$) $\rightarrow R, \boldsymbol{t}$

1. Determine the set of interest points within the calculated 2D contour $C$ of the object in the left camera image $I_l$.
2. For each calculated point, determine a correspondence in the right camera image $I_r$ by computing the *Zero Normalized Cross Correlation* (ZNCC) along the epipolar line.
3. Calculate a 3D point for each correspondence.
4. Fit a 3D model of the object into the calculated 3D point cloud and return the resulting rotation $R$ and the translation $\boldsymbol{t}$.





# Experimental Evaluation



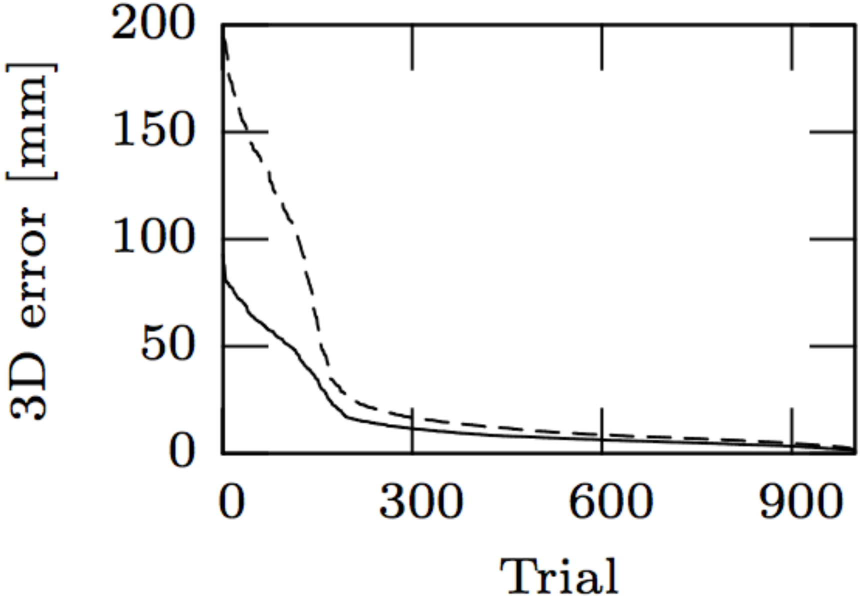

Solid line: Stereo-based
Dashed line: Monocular

the object of interest was moved along (resp. rotated around) a **single degree of freedom** for each plot

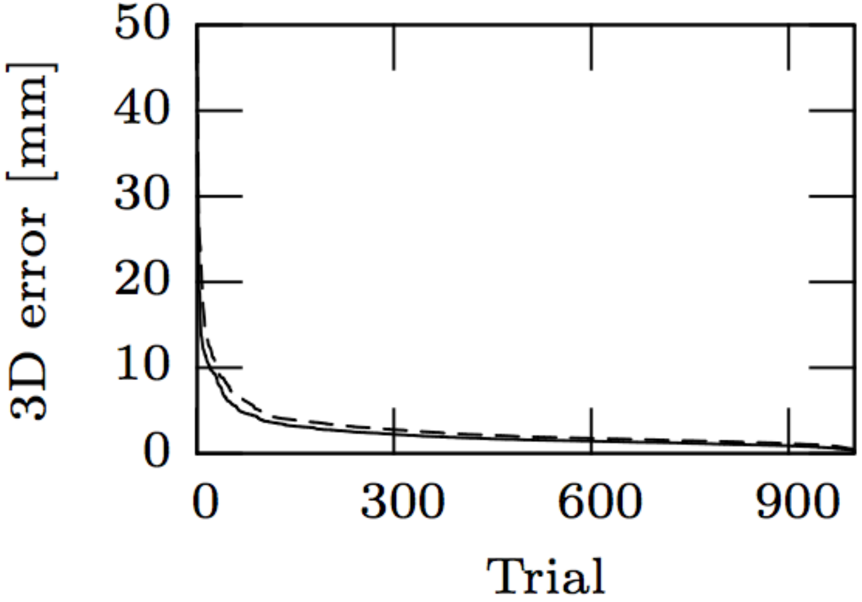
[1]

# Experimental Evaluation

Monocular




Stereo-Based




[1]

Solid line: Average error

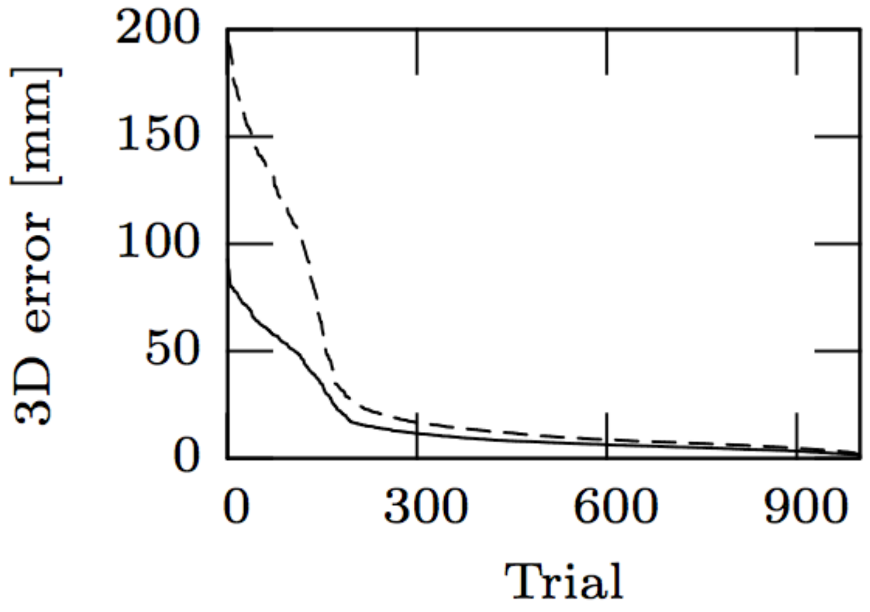
Dashed line: Maximum error

1,000 random poses were evaluated

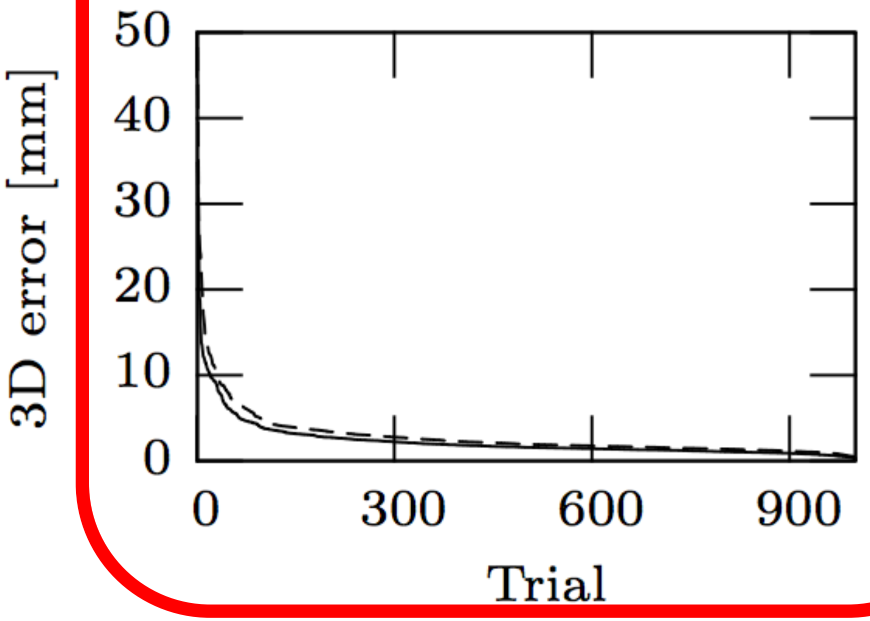
** The 3D error was measured on the basis of sampled 3D surface points.*

# Experimental Evaluation

Monocular




Stereo-Based




[1]

Solid line: Average error

Dashed line: Maximum error

1,000 random poses were evaluated

** The 3D error was measured on the basis of sampled 3D surface points.*

# Experimental Evaluation

[1]

Left: Monocular (instable)
Right: Stereo-based

# Experimental Evaluation

Standard deviation for the estimated pose of a static object (calculated for 100 frames)

| | $x$ | $y$ | $z$ | $\theta_x$ | $\theta_y$ | $\theta_z$ |
|---|---|---|---|---|---|---|
| Proposed method | 0.23 | 0.42 | 0.39 | 0.066 | 0.17 | 0.10 |
| Conventional method | 0.24 | 0.038 | 1.52 | 0.17 | 0.29 | 0.13 |

[1]

*Units in [mm] and [degrees].*

*Proposed Method = Stereo-based Approach*
*Conventional Method = Monocular Approach (only stable situations)*

# Experimental Evaluation

Standard deviation for the estimated pose of a static object (calculated for 100 frames)

| | $x$ | $y$ | $z$ | $\theta_x$ | $\theta_y$ | $\theta_z$ |
|---|---|---|---|---|---|---|
| Proposed method | 0.23 | 0.42 | 0.39 | 0.066 | 0.17 | 0.10 |
| Conventional method | 0.24 | 0.038 | 1.52 | 0.17 | 0.29 | 0.13 |

[1]

*Units in [mm] and [degrees].*

*Proposed Method = Stereo-based Approach*
*Conventional Method = Monocular Approach (only stable situations)*

Standard deviation of the z-coordinate amounts to 1.52 mm for monocular approach and 0.39 mm for stereo-based approach

# Assessment

Pros:

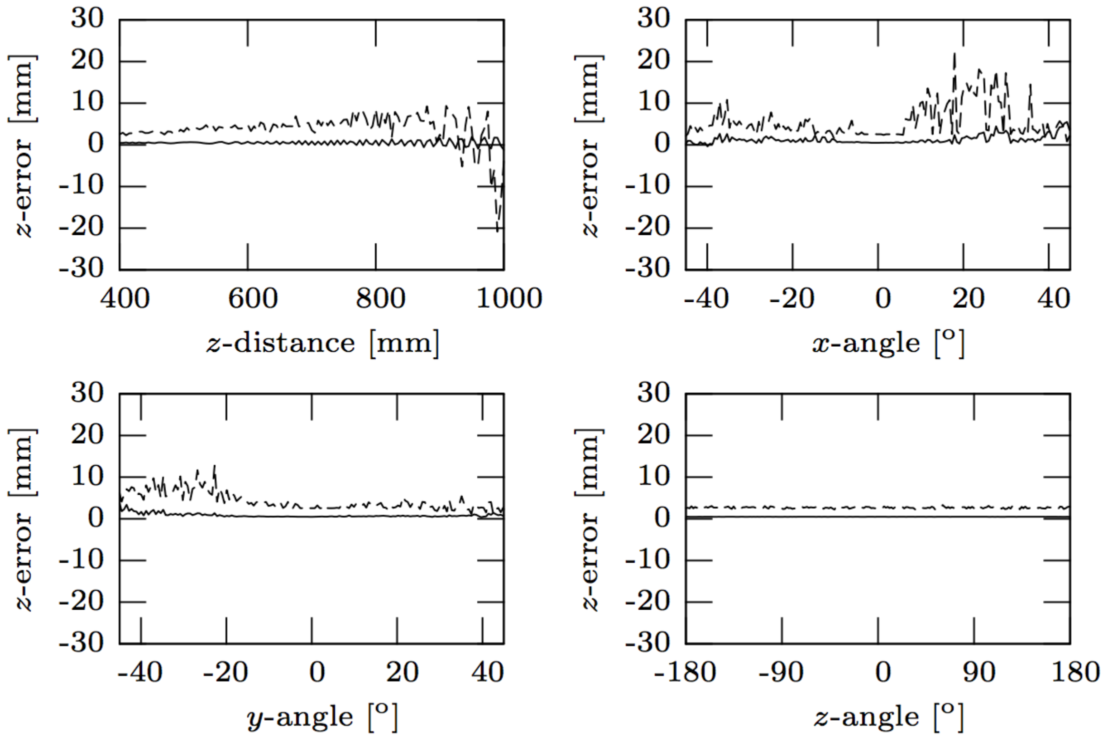
- Multiple tests of accuracy
  - Theoretical Accuracy Comparison and Experimental Evaluations
- Algorithm for Stereo-based Pose Estimation
  - Experimentally evaluated
- Detail about different toolkits (i.e. Integrating Vision Toolkit, IVT)
  - Compared the running time of different toolkits
    - Harris Corner Detector → Keyetech: 5ms, IVT: 10ms, OpenCV: 17ms

Cons:

- No detail on what kind of poses used for 1000 random pose evaluation
  - How many DoF were altered? To what extent the poses differed?

# Possible next step

- Tested for 1-DoF
  - How about 2-DoF, 3-DoF, so on so forth?
    - Translation AND Rotation?

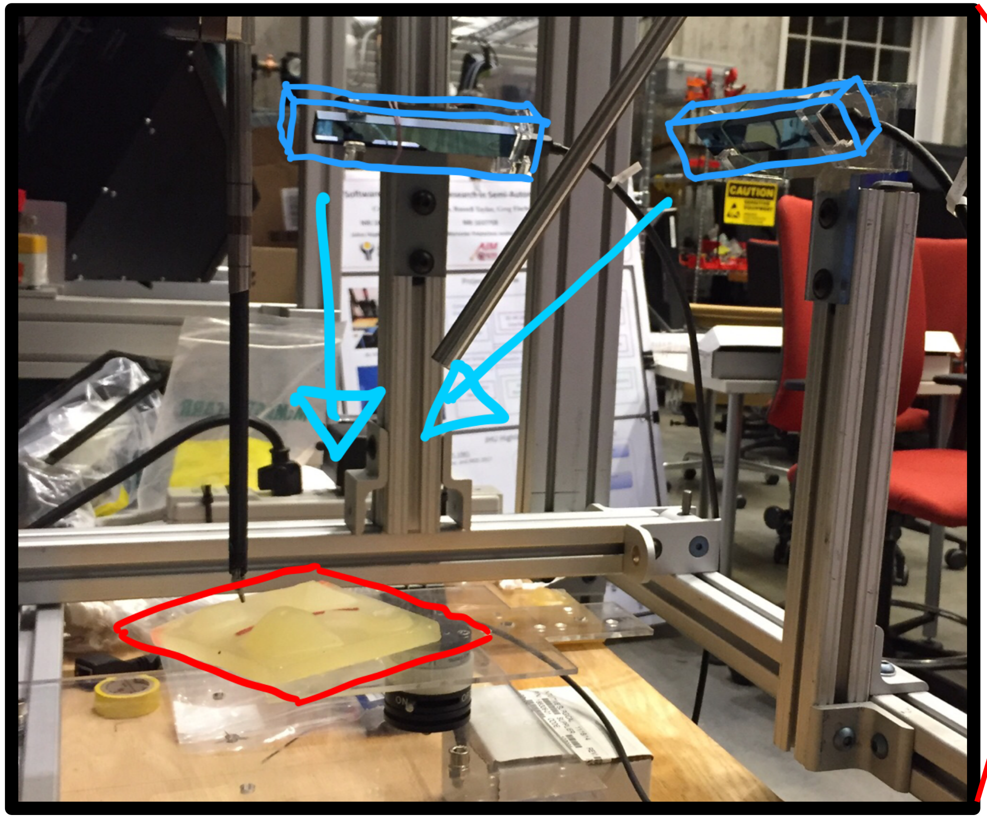


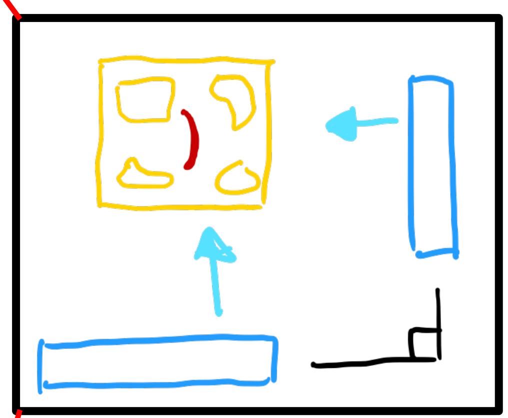

[1]

# Conclusion

Stereo Camera System has **higher accuracy** and **higher stability/robustness!**

→ And larger baseline = smaller error





# References

[1] Azad P., Asfour T., Dillmann R. “Stereo-Based vs. Monocular 6-DoF Pose Estimation Using Point Features: A Quantitative Comparison.” In: *Dillmann R., Beyerer J., Stiller C., Zöllner J.M., Gindele T. (eds) Autonome Mobile Systeme 2009.* Informatik aktuell. Springer, Berlin, Heidelberg

[2] P. Azad. *Visual Perception for Manipulation and Imitation in Humanoid Robots.* PhD thesis, Universit ̈at Karlsruhe (TH), Karlsruhe, Germany, 2008.

[3] P. Azad, T. Asfour, and R. Dillmann. "Combining Harris Interest Points and the SIFT Descriptor for Fast Scale-Invariant Object Recognition.” In *IEEE/RSJ In- ternational Conference on Intelligent Robots and Systems (IROS),* St. Louis, USA, 2009.

[4] P. Azad, T. Asfour, and R. Dillmann. Stereo-based 6D Object Localization for Grasping with Humanoid Robot Systems. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 919–924, San Diego, USA, 2007.