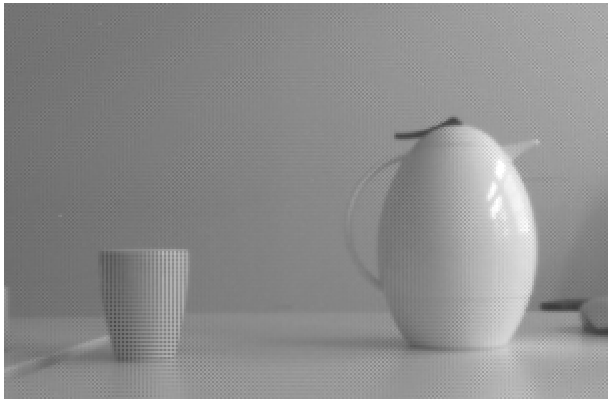
Detection of Household Objects Using Projected Light Patterns

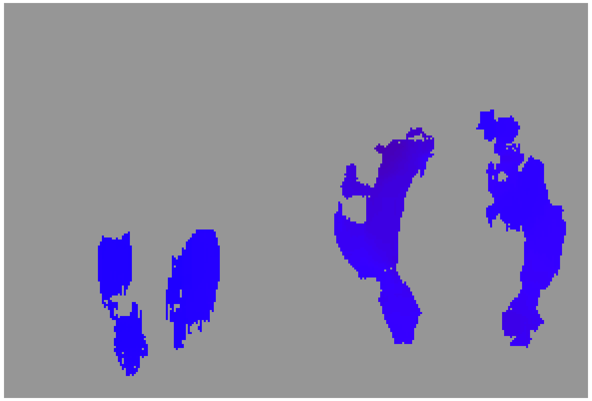
Advisor: Dejan Pangercic

Supervisor: Prof. Michael Beetz, Ph.D.

Motivation

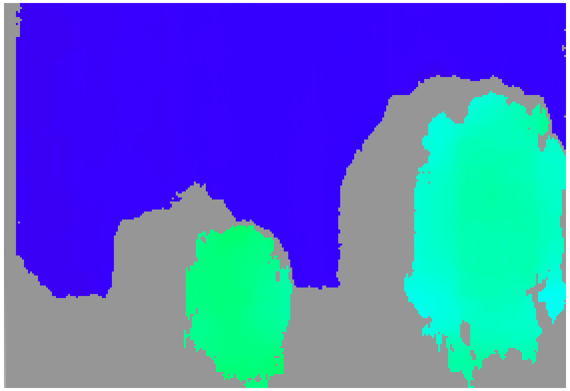
Stereo vision bases upon finding correspondences between two camera images





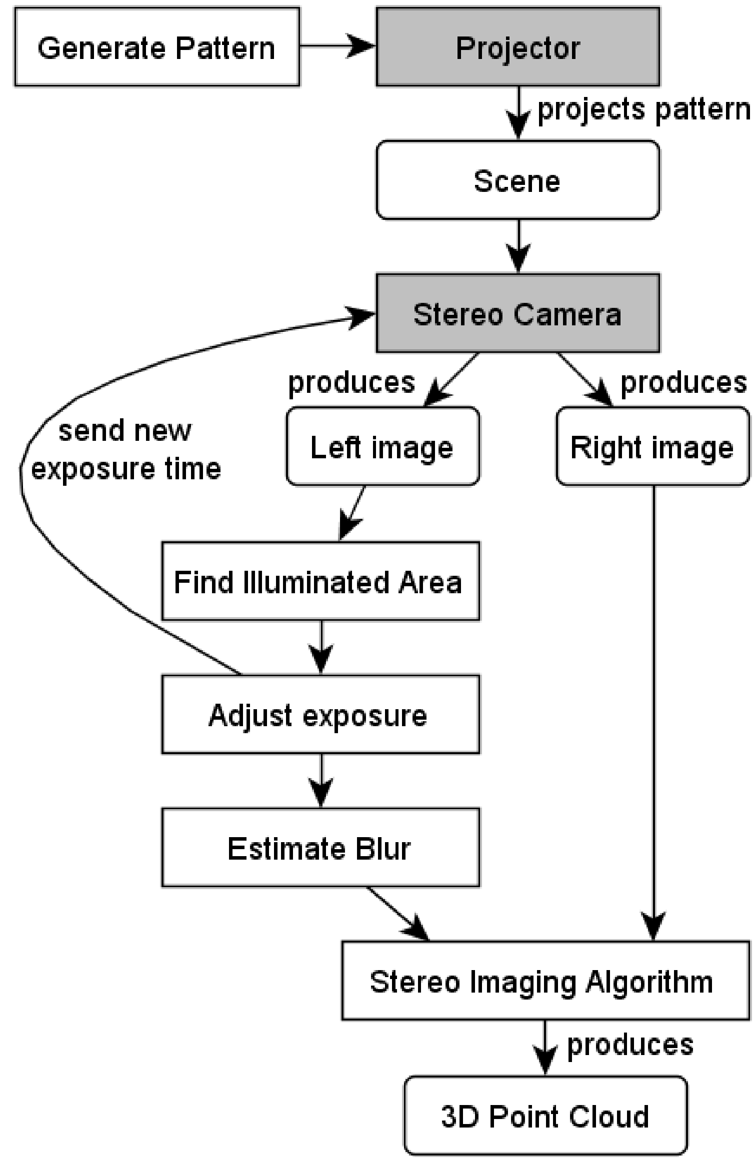
=> Impossible for objects with no texture

Possible solution:
Providing artificial texture by projecting a pattern on the scene



Outline

- **Finding the Illuminated Area**
- **Adjustment of Exposure Time**
- **Blur Estimation**
- **Comparison of Point Clouds**
- **Results**
- **Conclusion and Future Work**



Finding Illuminated Area

- Useful to reduce the image to the Region-Of-Interest
- Essential for the techniques presented in the next two chapters

Two different methods:

- Difference Image
- Background Subtraction

Finding Illuminated Area

Difference Image

Idea:

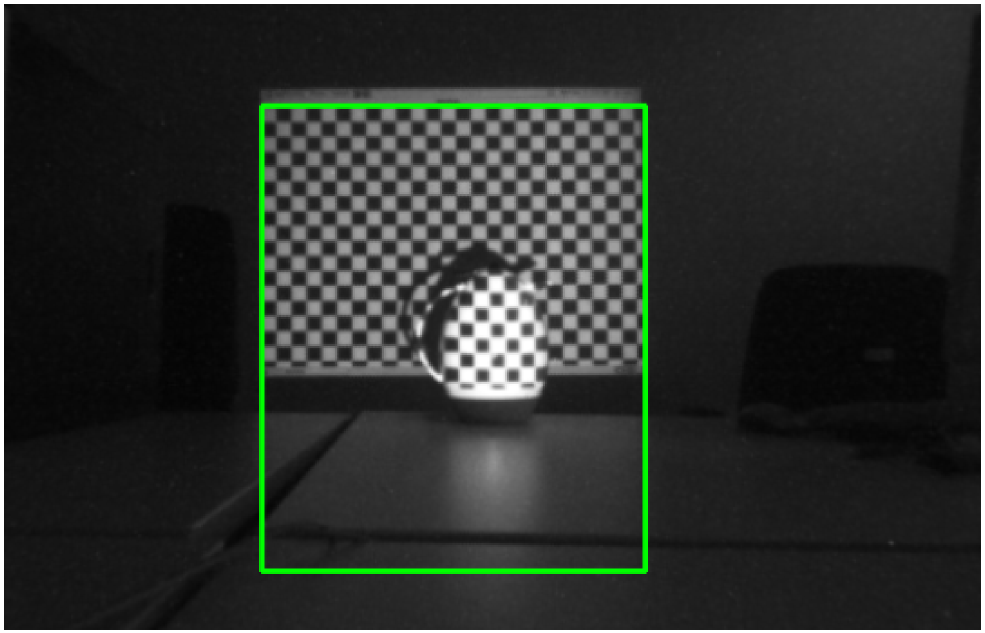
Two images: One with a white and one with a black projected pattern

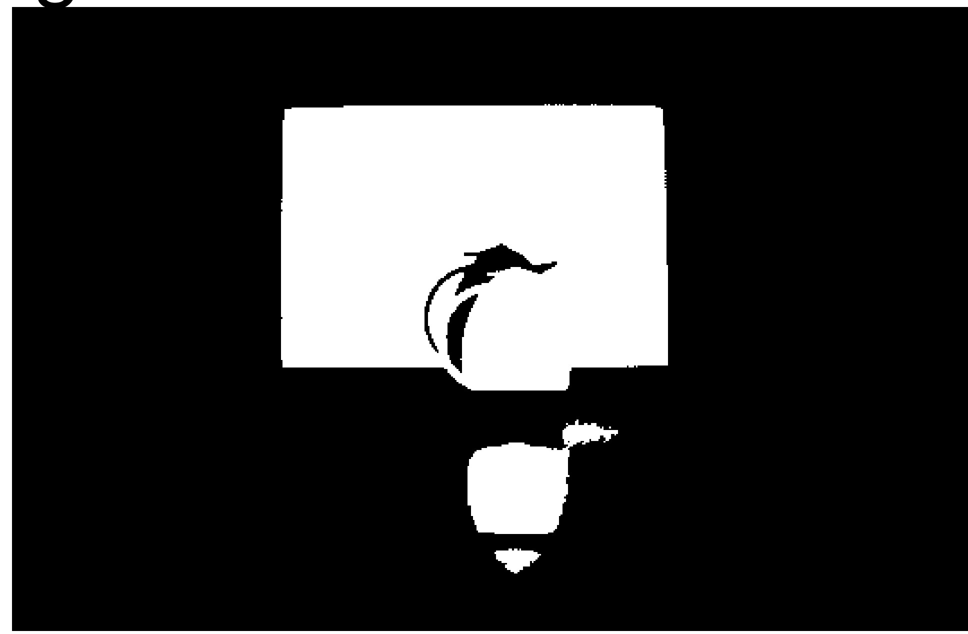
Difference = Illuminated Area

Ignore pixels whose difference is smaller than a certain percentage of the maximal difference to filter out reflections

Finding Illuminated Area

Problem: Strong reflections





Possible Solution:

Partitioning of the Difference Image

But: Expensive & works only if the reflections are separated from the rest

Finding Illuminated Area

Background Subtraction:

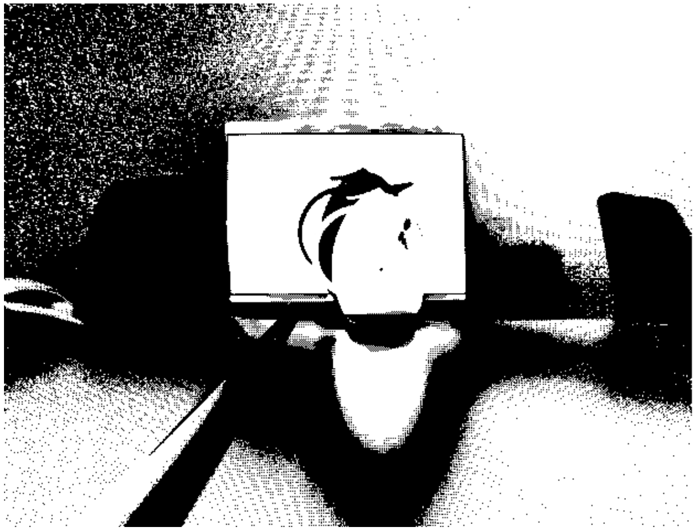
Learning a background with a black pattern

Finding the illuminated area as a foreground of an image with a projected pattern

Finding Illuminated Area

Used implementation: Background Subtraction with Codebooks

=> Finding the right parameters to filter out the reflections from the foregound image



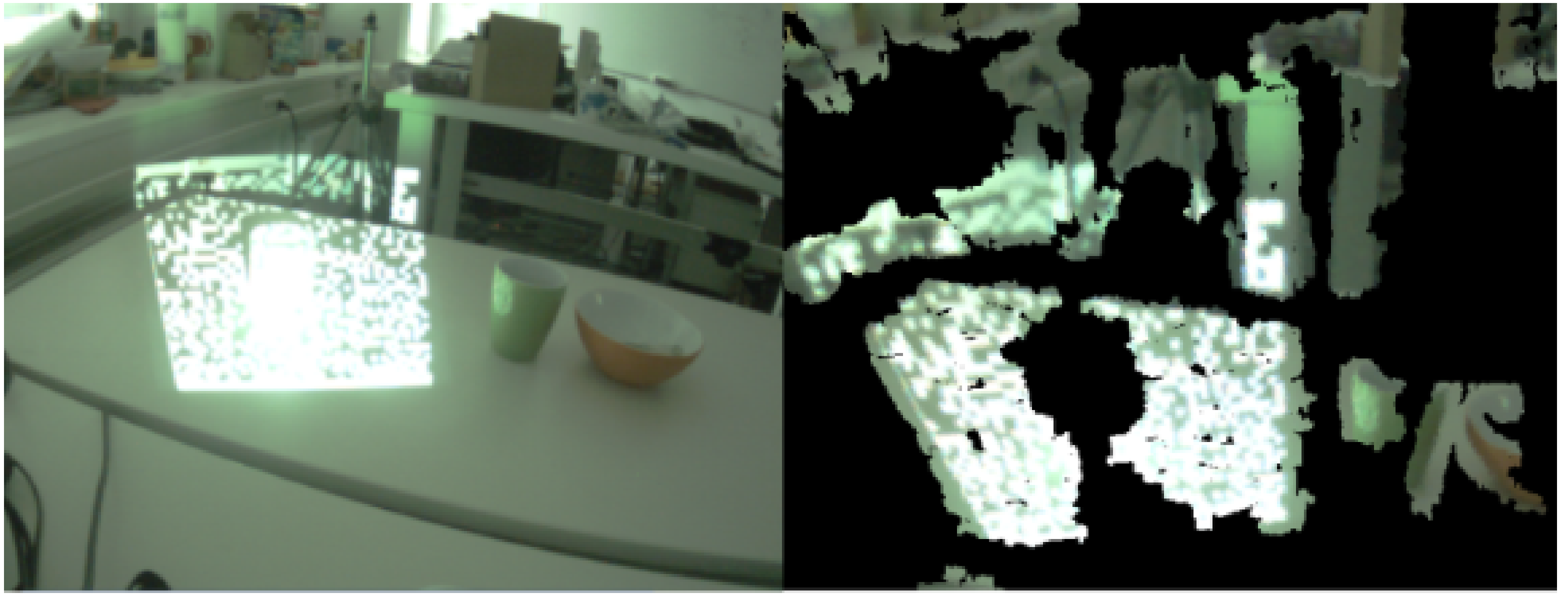
Default parameters

Adapted parameters

Adjustment of Exposure Time

A camera that automatically adjusts its exposure time tends to overexposes the part of the image in which the pattern is projected

=>Poor resulting point clouds



Camera image | Point cloud

Adjustment of Exposure Time

Shorter exposure time =>

Much better results in the illuminated area,
but rest of the scene is very dark

Camera image

Point cloud

Adjustment of Exposure Time

Idea:

Average brightness in the illuminated area

=

Constant that is used by the automatic exposure adjustment for the whole image

$$B_{opt,proj} = \overline{B_{mom}} + B_{opt,auto} - \overline{B_{ROI}}$$

Linear relation between exposure time and average brightness:

$$exp_{opt} = \frac{B_{opt,proj}}{B_{mom}} exp_{mom}$$

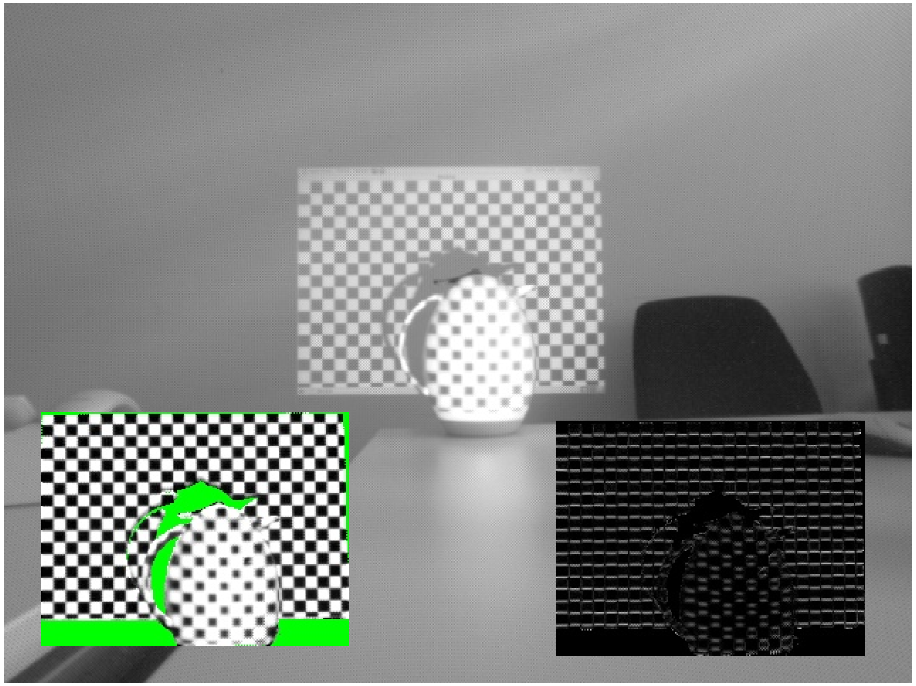
Blur Estimation

Blur estimation method of paper

„Light pattern blur estimation for automatic focus control of structured light 3d camera“
(Lam Quang Bui, The 2009 IEEE/RSJ International Conference onIntelligent Robots and Systems)

Main Idea:

- Normalize Image
- Estimate derivation
- Model edges with a Gaussian Function



- Standard deviation σ describes the amount of blur

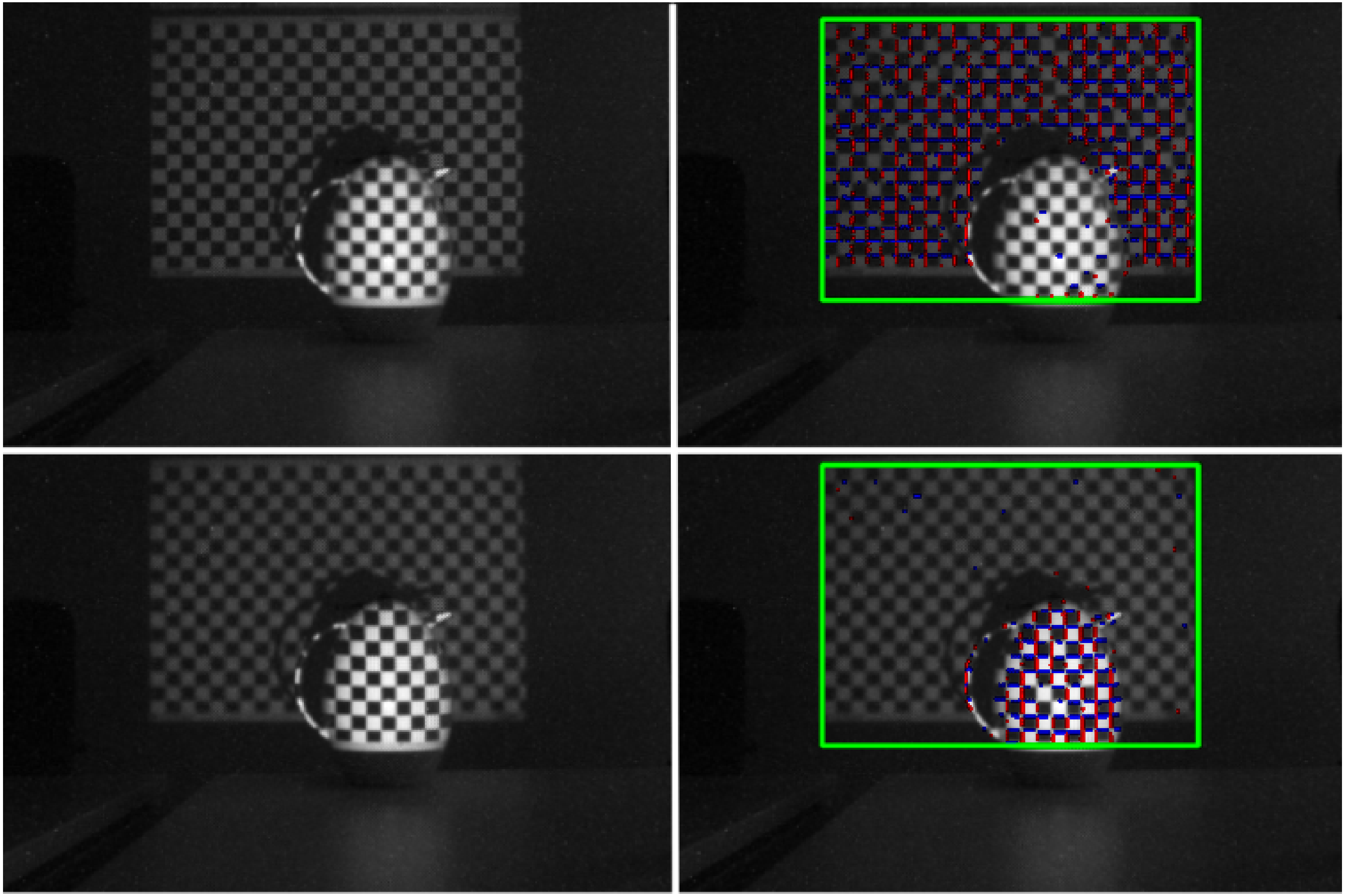
Blur Estimation

Modifications of the proposed method:

- Filter out dark pixels within the illuminated area
  =>Reduces noise in that regions

- No searching for lines, but estimating the blur in every pixel
  => Other patterns than a stripe pattern can be used

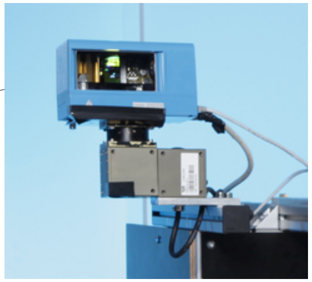
Blur Estimation

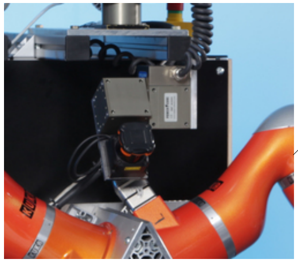


Comparison of Different Point Clouds

Data sources:

- Projected stereo with different patterns
- Virtual scanner
- SICK LMS 400
- Hokuyo UTM 30LX



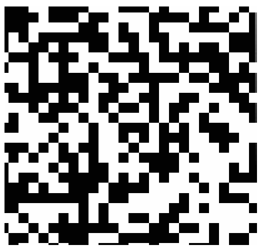


Data evalution methods:

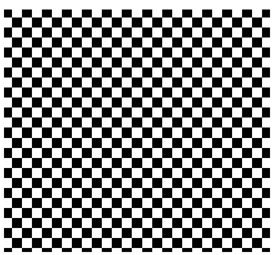
- Pixel Density
- ICP fitness score

Comparison of Different Point Clouds

Different patterns:



Random pattern

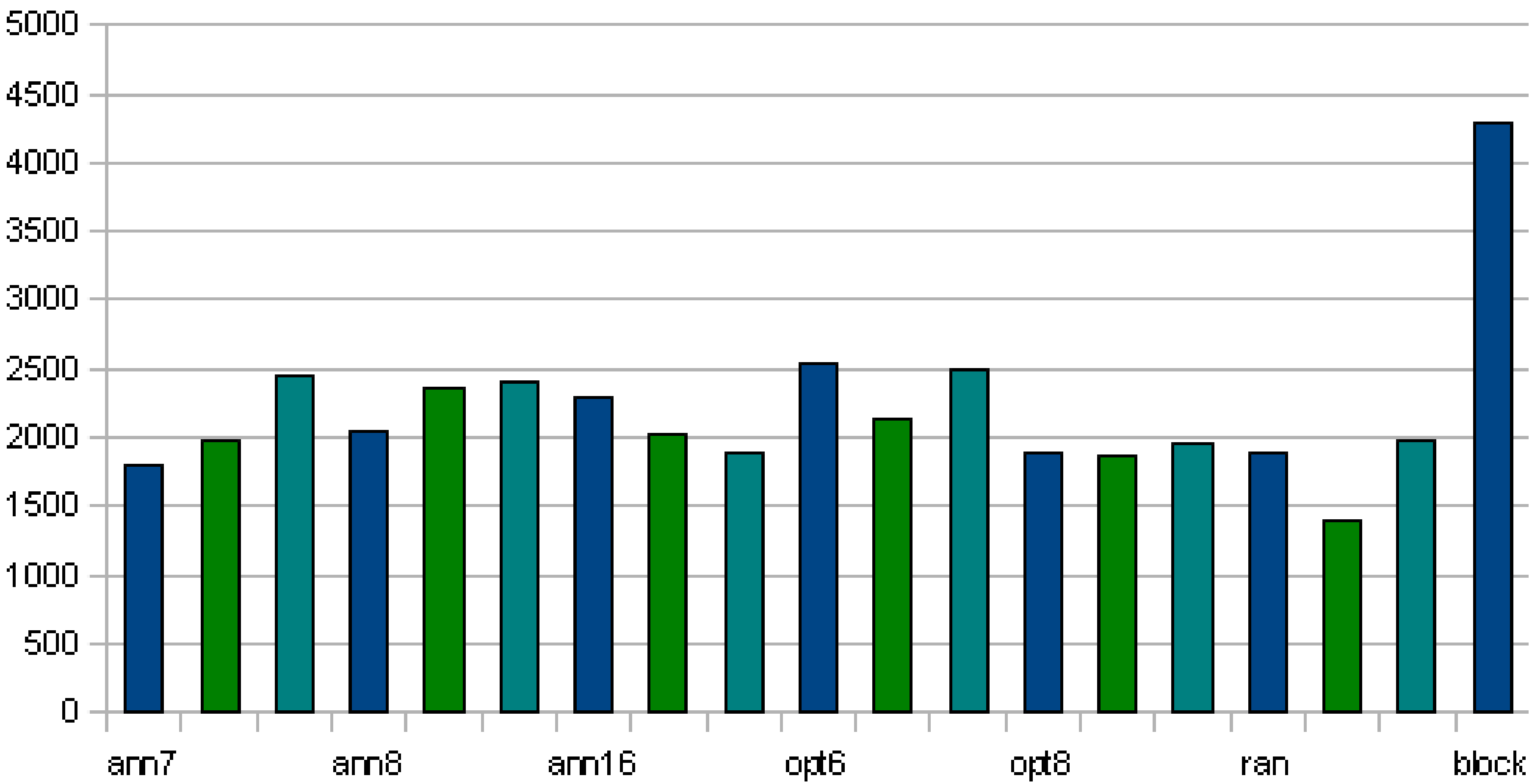


Block pattern

Different patterns created with simulated annealing
(based on Kurt Konolige's paper „Projected texture stereo, ICRA 2010)

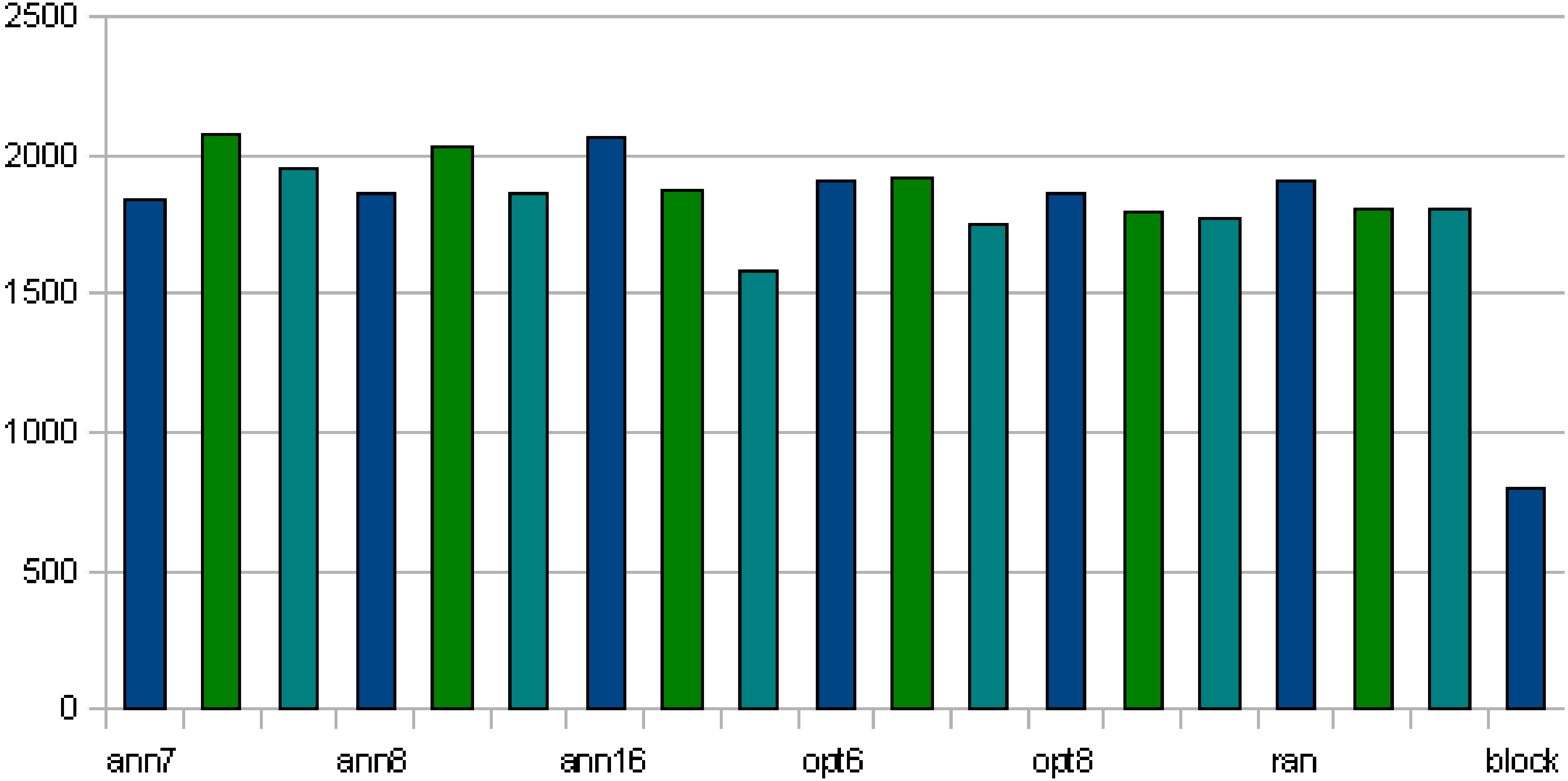
Results – Different Patterns

ICP Fitness Score



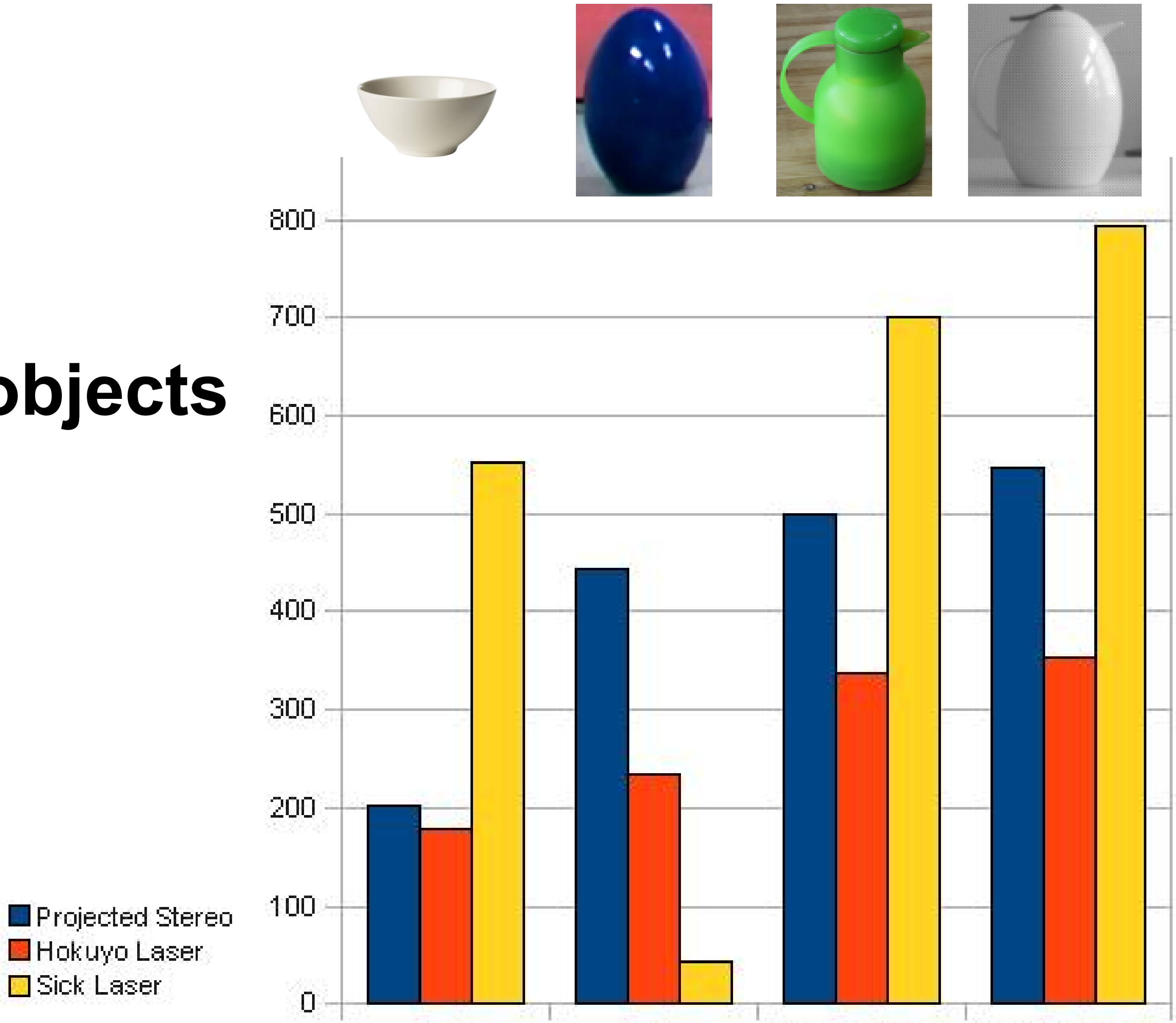
Results – Different Patterns

Number of Points



Results – Comparison with Laser

Pixel Density for different objects



Conclusions and Future Work

Ideas for Future Work

- Blur can only be estimated, but not adjusted
- Hardware system for focus control would be needed
- Consideration of color, too, to improve the detection of objects

Conclusion and Future Work

ROS nodes that implement all introduced methods can be found in the `ias_projected_light` package at `www.ros.org`

A tutorial at:

www.ros.org/wiki/ias_projected_light