Detection and Fine 3D Pose Estimation of Texture-less Objects in RGB-D Images

Tomáš Hodaň[1], Xenophon Zabulis[2], Manolis Lourakis[2], Štěpán Obdržálek[1], Jiří Matas[1]

[1] Center for Machine Perception, CTU in Prague, CZ

[2] Institute of Computer Science, FORTH, Heraklion, GR

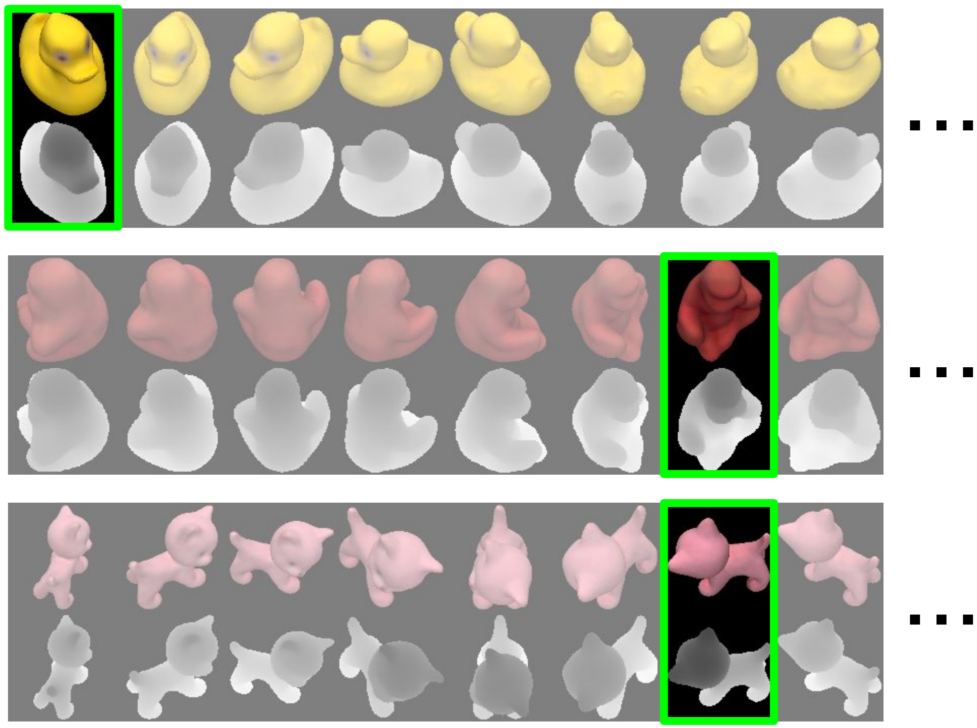
1st October 2015, Hamburg

Texture-less Objects in Robotics

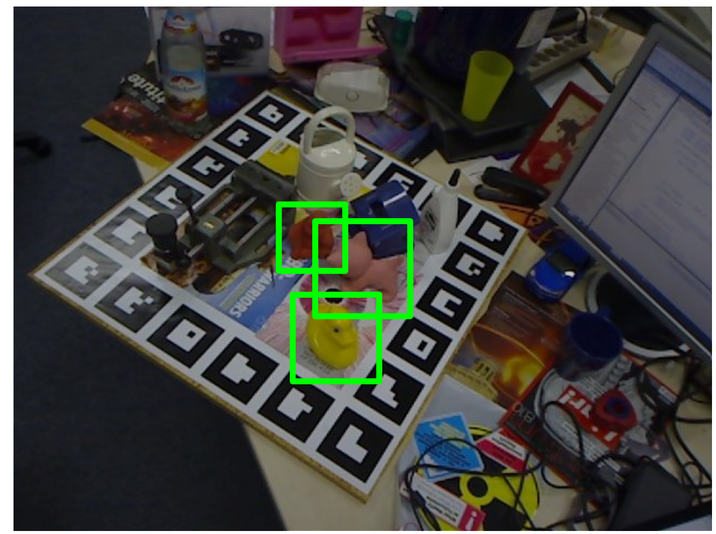
Detection and accurate localization of texture-less objects is commonly required in personal and industrial robotics

Problem Formulation

Given a database of training RGB-D images annotated with 3D poses, **detect all instances of known objects** in a test image and **estimate their 3D poses**

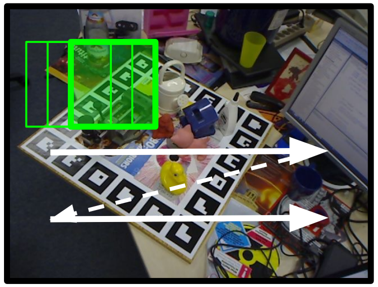


Training RGB-D images annotated with 3D poses

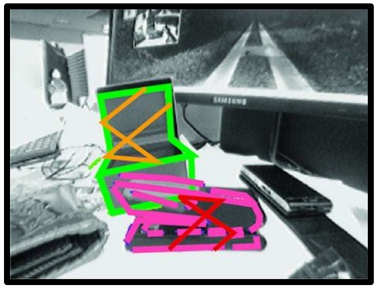


Test RGB-D image

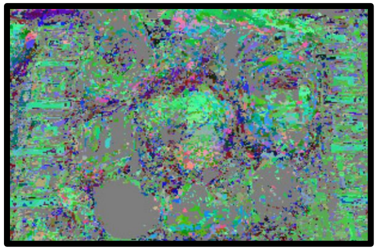
Related Work



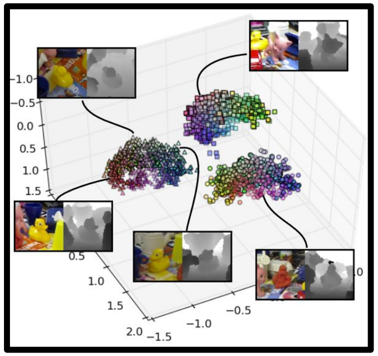
1. **Template matching methods**
   Hinterstoisser (ICCV 2011), Rios-Cabrera (ICCV 2013),
   Cai (ICVS 2013)



2. **Shape matching methods**
   Damen (BMVC 2012), Tombari (ICCV 2013), Drost (CVPR 2010),
   Choi (IROS 2012)



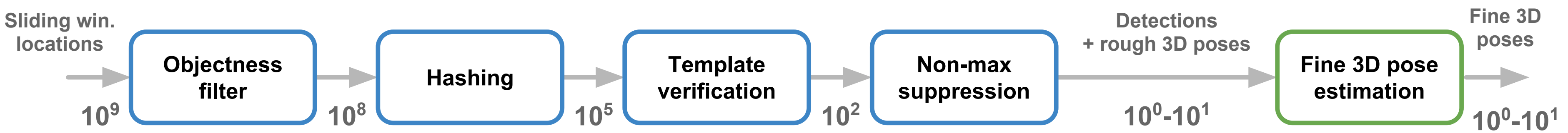
3. **Methods based on dense features**
   Sun (ECCV 2010), Gall (PAMI 2011), Brachmann (ECCV 2014)



4. **Deep learning methods**
   Wohlhart (CVPR 2015), Held (arXiv 2015), Krull (arXiv 2015)

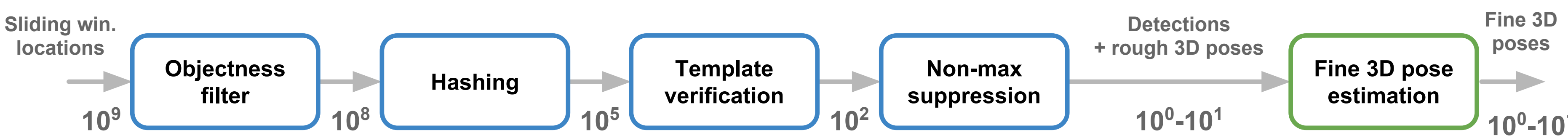
The Proposed Method

- Multi-scale **sliding window**
- **Efficient cascade-style evaluation** of each location
- The window has a **fixed size**, the same as the templates
- Stochastic optimization used to **refine the 3D pose**



The Proposed Method

- Multi-scale **sliding window**
- **Efficient cascade-style evaluation** of each location
- The window has a **fixed size**, the same as the templates
- Stochastic optimization used to **refine the 3D pose**



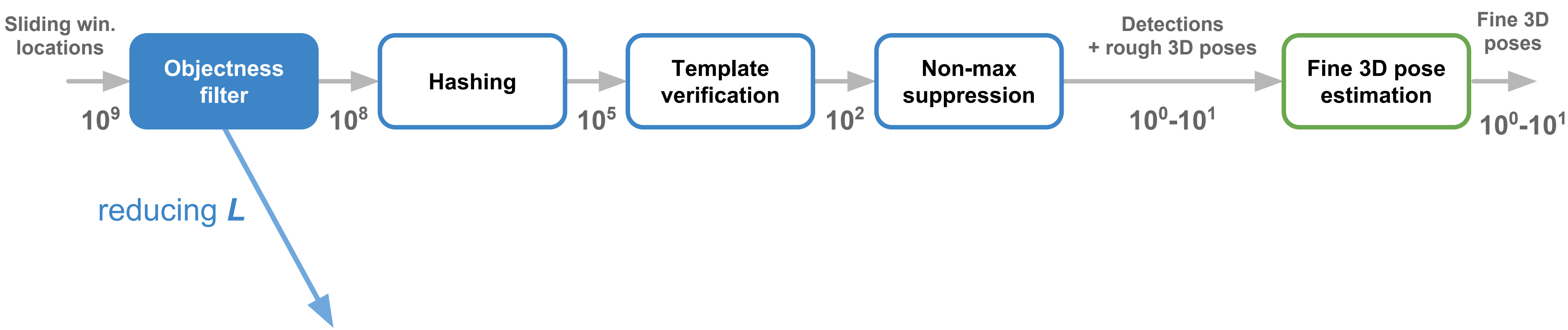
$O(LT)$ = complexity of an exhaustive template matching

L = the number of **sliding window locations**

T = the number of **training templates**

The Proposed Method

- Multi-scale **sliding window**
- **Efficient cascade-style evaluation** of each location
- The window has a **fixed size**, the same as the templates
- Stochastic optimization used to **refine the 3D pose**



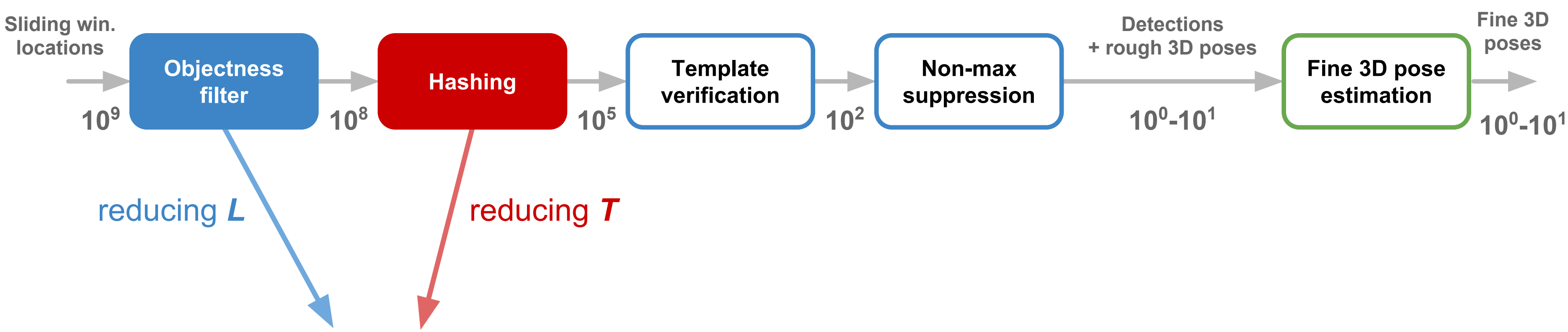
$O(LT)$ = complexity of an exhaustive template matching

L = the number of **sliding window locations**

T = the number of **training templates**

The Proposed Method

- Multi-scale **sliding window**
- **Efficient cascade-style evaluation** of each location
- The window has a **fixed size**, the same as the templates
- Stochastic optimization used to **refine the 3D pose**

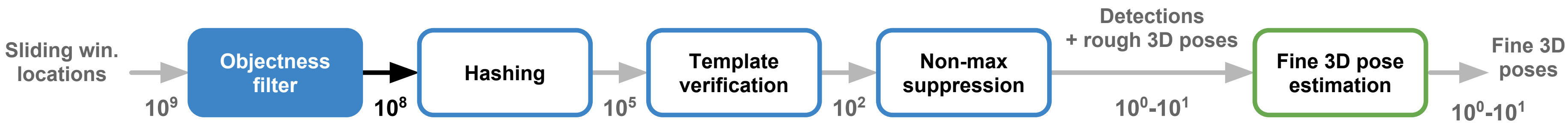


$O(LT)$ = complexity of an exhaustive template matching

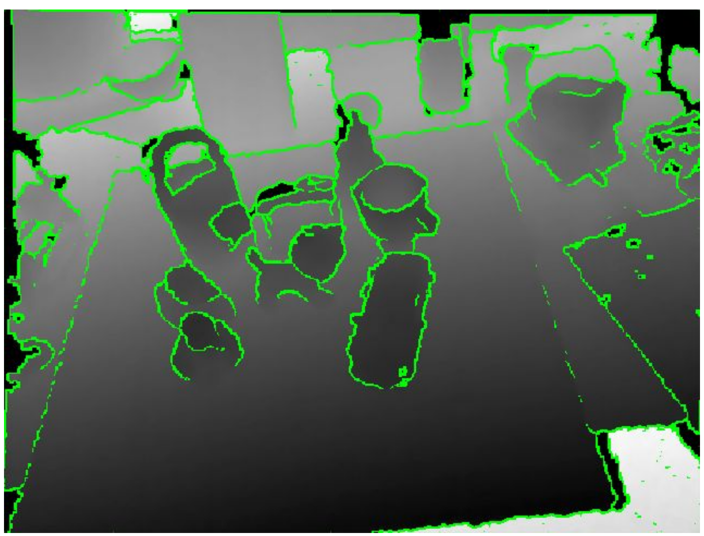
L = the number of **sliding window locations**

T = the number of **training templates**

Objectness Filter

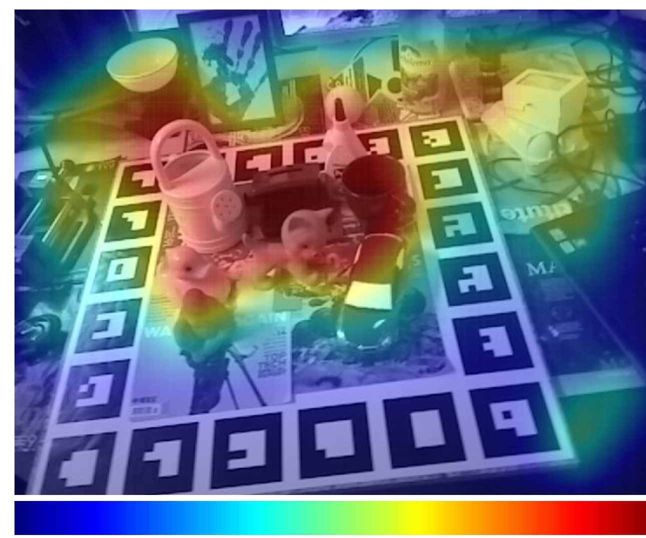


- Based on the **number of depth edges**
- The number of depth edges in a window is required to be **at least 30% of the minimum from the training templates**
- For false negative rate = 0, **60-90% of locations are pruned**
- Other window proposal methods (e.g. Edge-boxes) are being considered



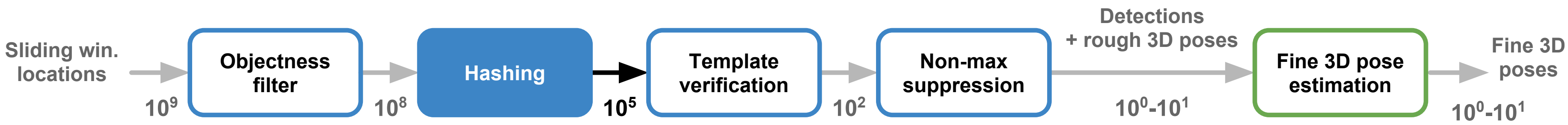
Detected depth edges

Number of detection candidates: 1.7 x 10^8



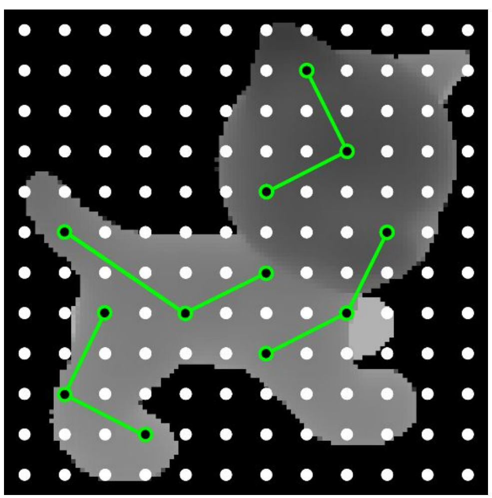
Density of detection candidates
detection candidate = (tpl. id, x, y, scale)

Hashing

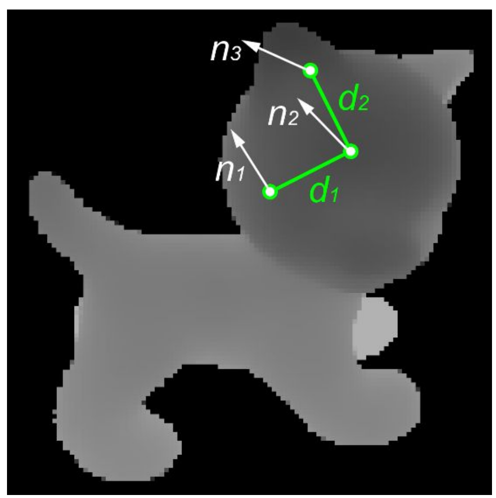


- Voting procedure based on **hashing descriptors of trained triplets of reference points** located on a grid
- Each triplet is described by **surface normals and depth differences**
- **Up to N templates with the most votes** are selected per location

Typically: N = 100, 8 bins for surface normal orientation, 5 bins for depth difference, i.e. $5^2 8^3$ = 12800 hash table bins

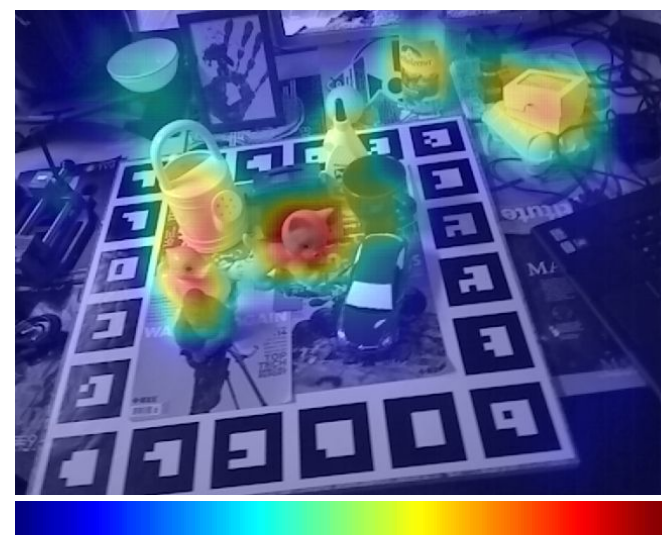


Sample triplets



Triplet description

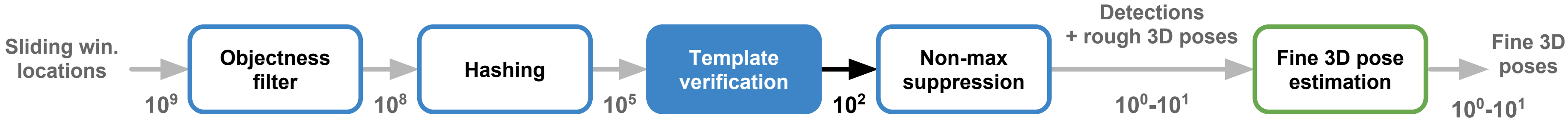
Number of detection candidates: 5.2 x 10^5



Density of detection candidates

detection candidate = (tpl. id, x, y, scale)

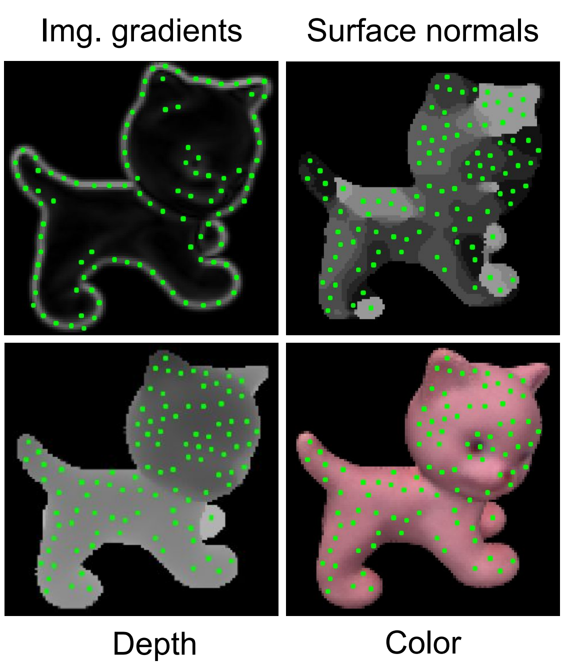
Multimodal Template Verification



- **A sequence of tests** evaluating consistency of:
  a. Object size and the measured depth
  b. Surface normals
  c. Image gradients
  d. Depth
  e. Color (HSV)

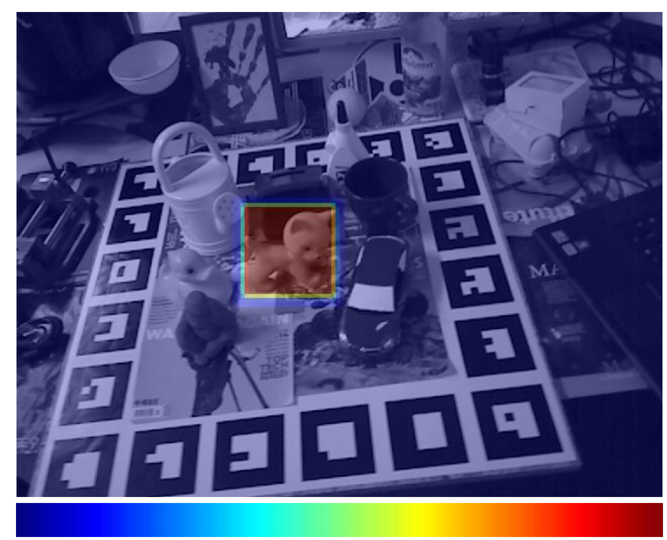
Evaluated on learnt feature points

Based on: Hinterstoisser et al., "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes", ICCV, 2011



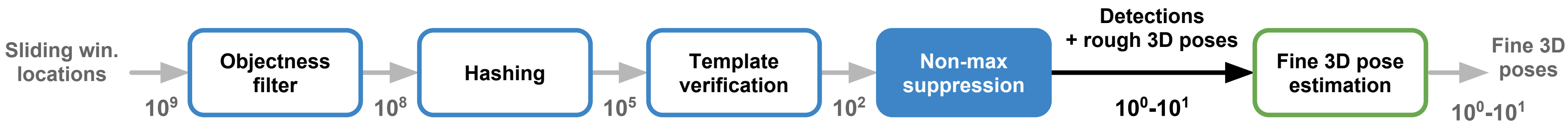
Learnt feature points in different modalities

Number of detections: 44



Density of detection candidates
detection candidate = (tpl. id, x, y, scale)

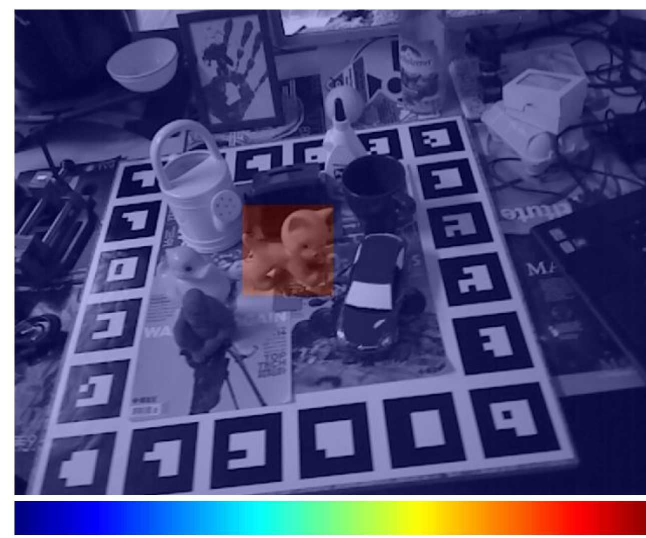
Non-maxima Suppression



- Detection candidates with **locally highest score are retained**
- The 3D poses associated with the detected templates are used as **initial poses** in the pose refinement procedure

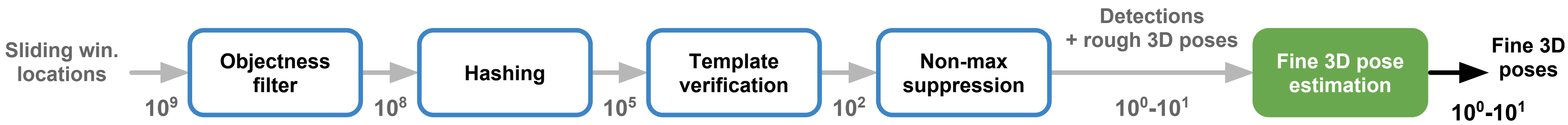
Rendering of the 3D pose associated with the detected template

Number of detections: 1



Density of detection candidates
detection candidate = (tpl. id, x, y, scale)

Fine 3D Pose Estimation



- The rough initial 3D pose is refined using a hypothesize and test scheme based on **Particle Swarm Optimization** (PSO)
- PSO stochastically evolves a population of candidate poses over multiple iterations
- Candidate poses are evaluated by comparing their rendered depth images to the input depth image (using a cost function measuring similarity in **depth, surface normals and depth edges**)
- Pose refinement using PSO is **less sensitive to local minima compared to ICP**

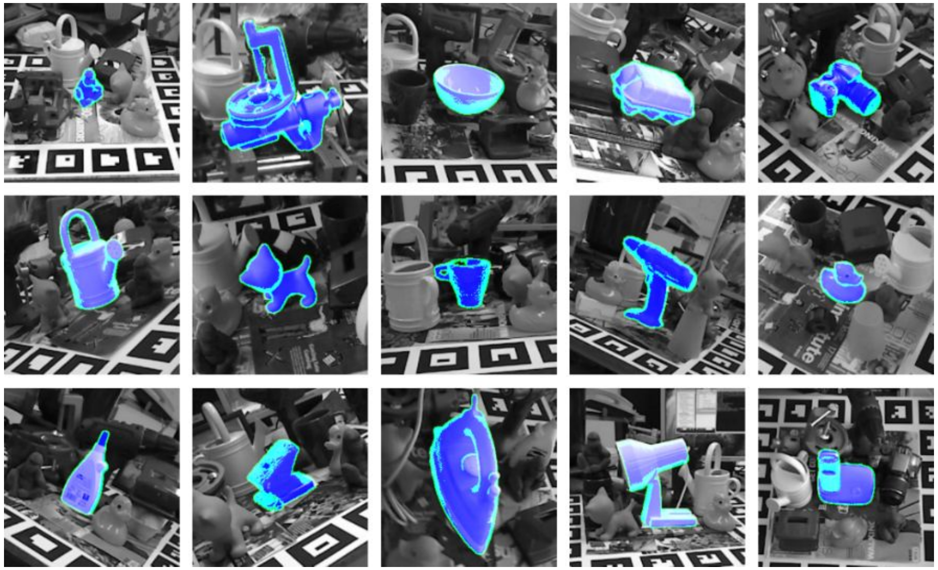
Details in: Zabulis, Lourakis and Koutlemanis, ”3D Object Pose Refinement in Range Images”, Intl Conf. on Computer Vision Systems, ICVS, 2015

Recognition Rate

- Evaluation on the **dataset of Hinterstoisser** [1]:
  - 15 texture-less objects, 1200 RGB-D test images for each
  - **Object localization**: detect the given object and estimate its pose
- The recognition rate (recall) of our method is **comparable to SOTA**

Sequence	Our method	LINEMOD++	LINEMOD	Drost et al.
1. Ape	93.9	**95.8**	69.4	86.5
2. Benchvise	**99.8**	98.7	94.0	70.7
3. Bowl	98.8	**99.9**	99.5	95.7
4. Box	**100.0**	99.8	99.1	97.0
5. Cam	95.5	**97.5**	79.5	78.6
6. Can	**95.9**	95.4	79.5	80.2
7. Cat	98.2	**99.3**	88.2	85.4
8. Cup	**99.5**	97.1	80.7	68.4
9. Driller	**94.1**	93.6	81.3	87.3
10. Duck	94.3	**95.9**	75.9	46.0
11. Glue	**98.0**	91.8	64.3	57.2
12. Hole punch	88.0	**95.9**	78.4	77.4
13. Iron	97.0	**97.5**	88.8	84.9
14. Lamp	88.8	**97.7**	89.8	93.3
15. Phone	89.4	**93.3**	77.8	80.7
Average	95.4	**96.6**	83.0	79.3

Recognition rates [%]
(LINEMOD and LINEMOD++ are methods from [1])

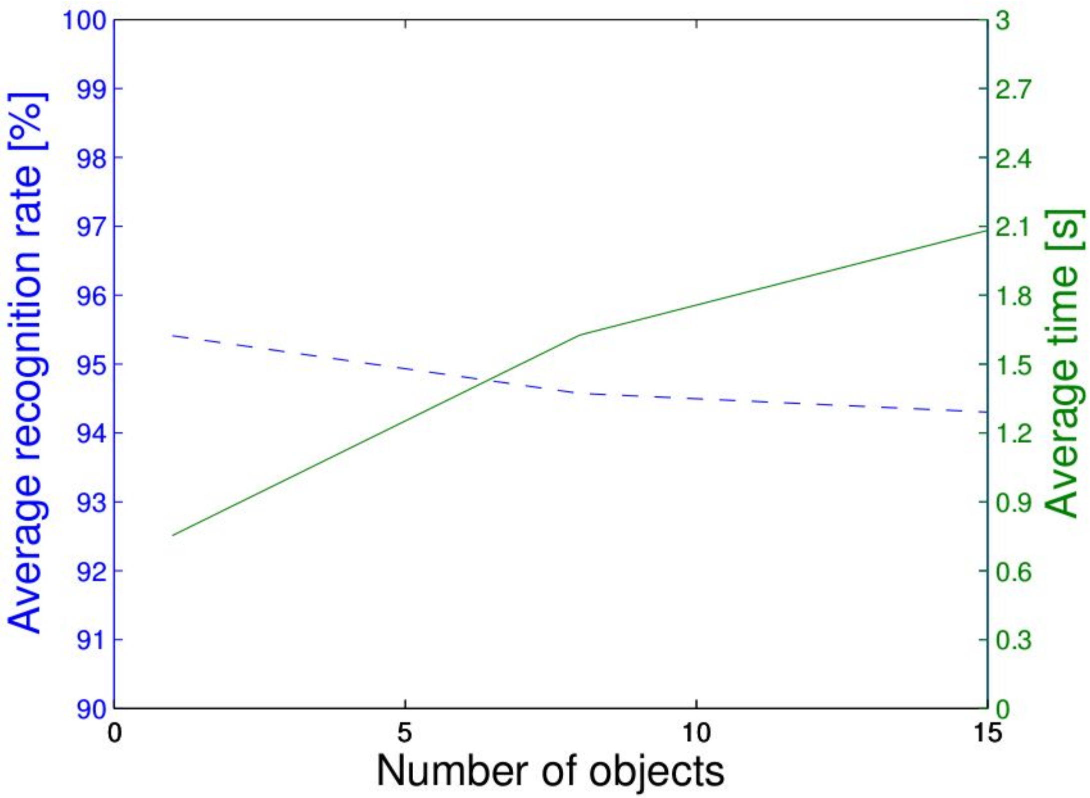


Sample 3D pose estimations

[1] Hinterstoisser et al., "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," ACCV, 2012
[2] Drost et al., "Model globally, match locally: Efficient and robust 3d object recognition," CVPR, 2010

Scalability and Speed

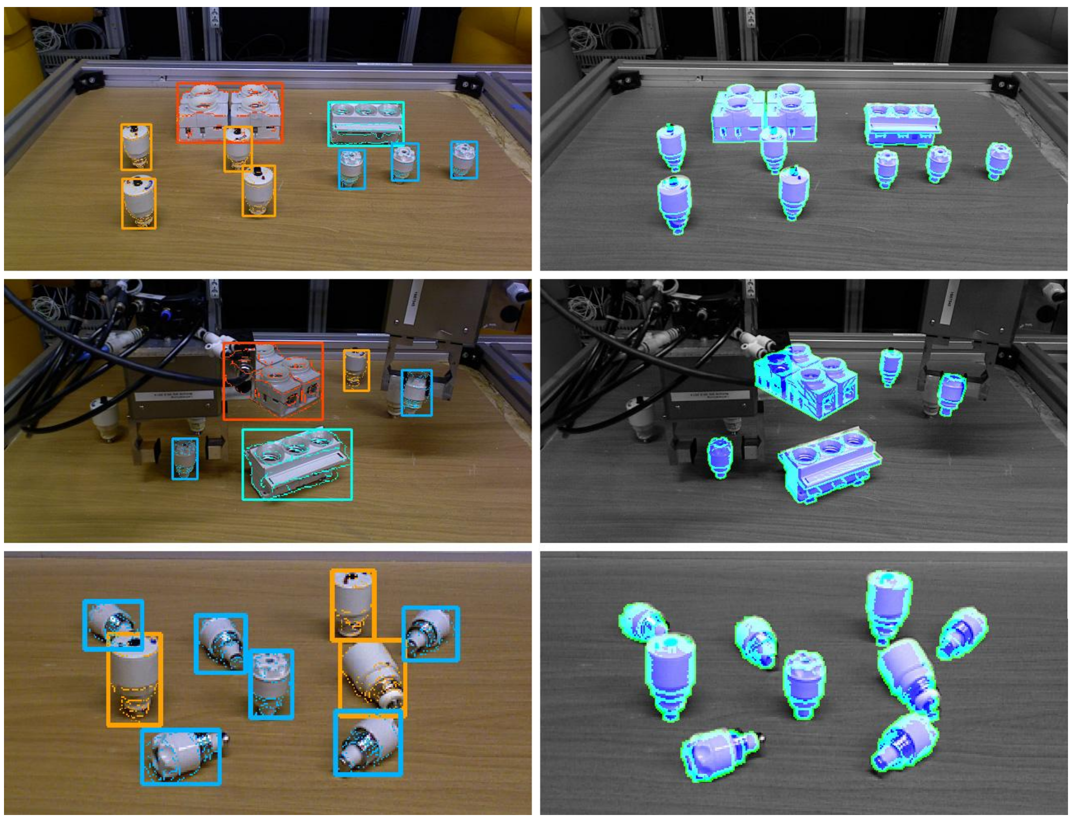
- Time complexity is **sub-linear in the number of templates**
- When the number of loaded templates increased 15 times, the average recognition time increased only less than 3 times:

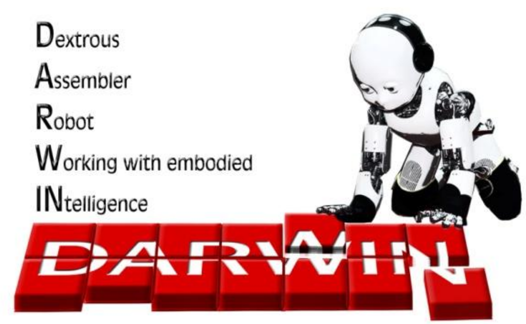


- **0.75 s** per VGA frame (9 image scales) for a single known object

Robotic Assembly Application

- An arm with a gripper is assigned the task of picking up electrical fuses at arbitrary locations in its workspace and inserting them into the sockets of corresponding fuse boxes
- **Detection and fine 3D pose estimation is crucial for this task**





Thank you!