The Object Recognition Kitchen

Vincent Rabaud

May 10, 2013

Warning

THIS IS NOT A TALK ABOUT YET ANOTHER OBJECT RECOGNITION METHOD !

Outline

1. Problematic
   - Previous Work
   - A Step Back
2. A Solution
   - Ecto
   - Database
   - Pipelines
   - Misc
3. Future Work
   - Better Pipelines
   - More Pipelines
   - More Integration
4. Conclusion

Outline

1. Problematic
Previous Work
A Step Back

2. A Solution
Ecto
Database
Pipelines
Misc

3. Future Work
Better Pipelines
More Pipelines
More Integration

4. Conclusion

Previous Work (1/2)

Techniques:

- BIG
- LARK
- CVFH
- TOD (Textured Object Detection)
- LINEMOD
- ...

Frameworks:

- objects of daily use (TUM)
- MOPED
- RoboEarth
- ...

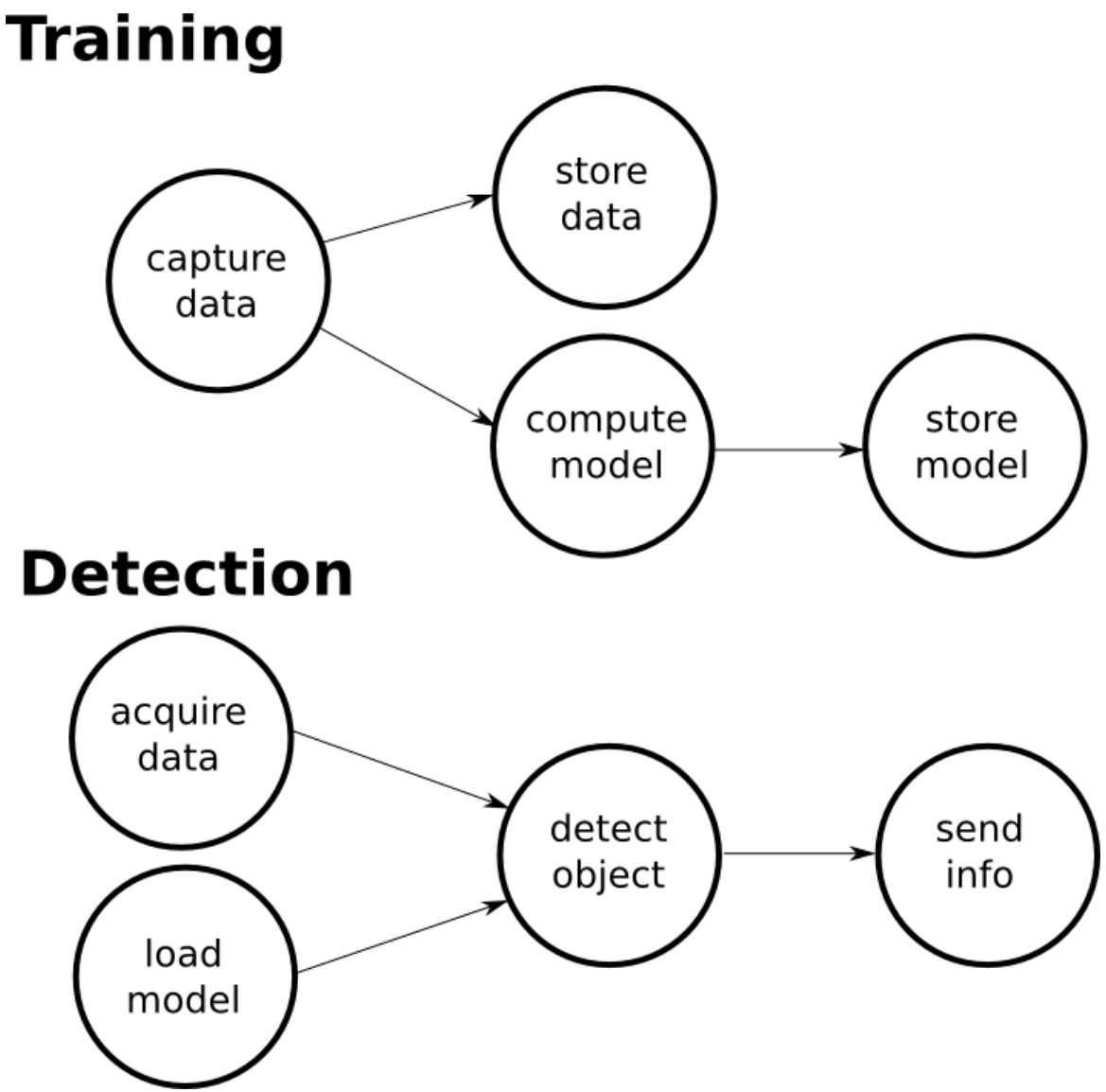
Is there really a problem ?

Previous Work (2/2)

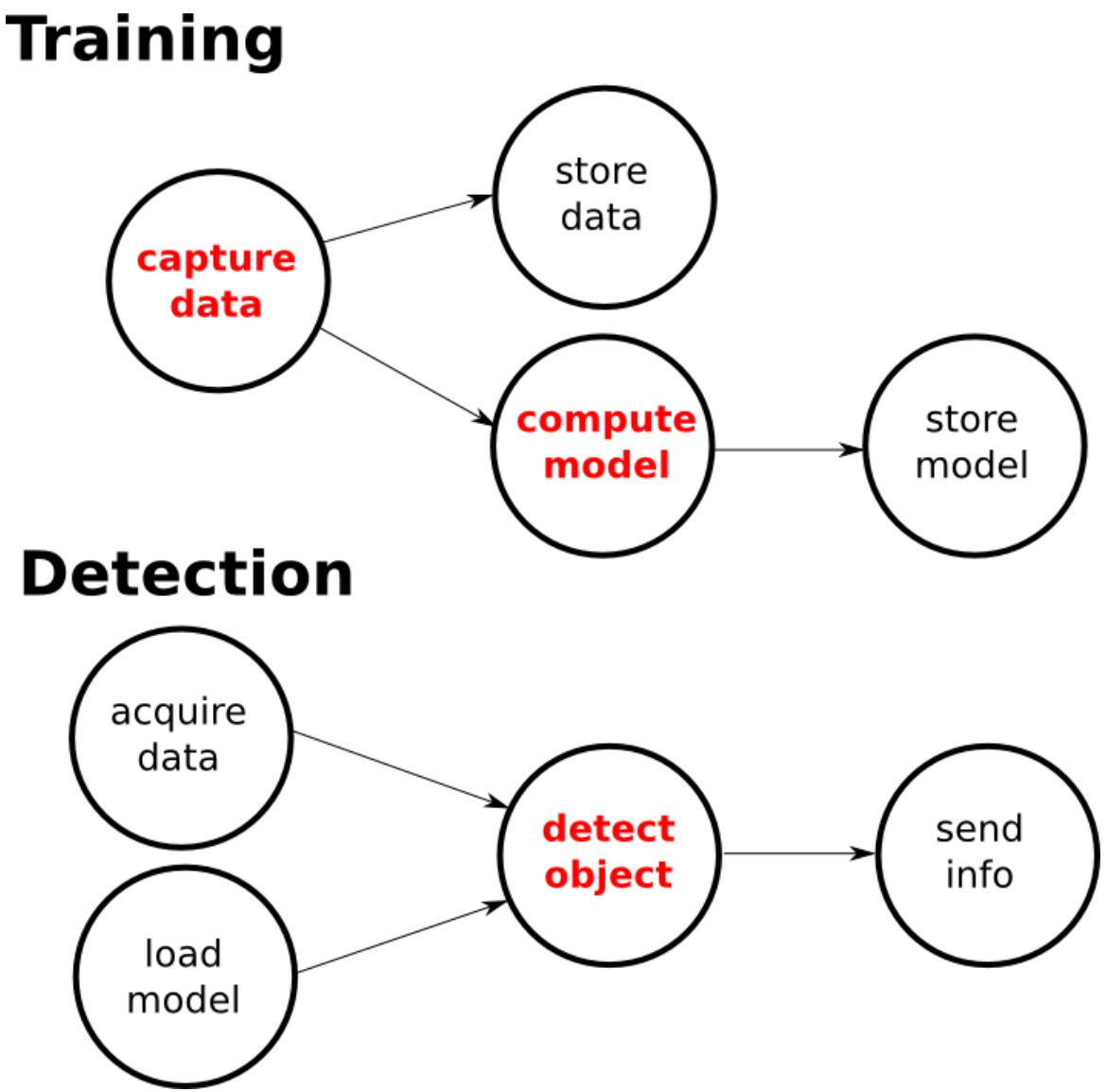
Individually, not really. E.g.

LINEMOD Video

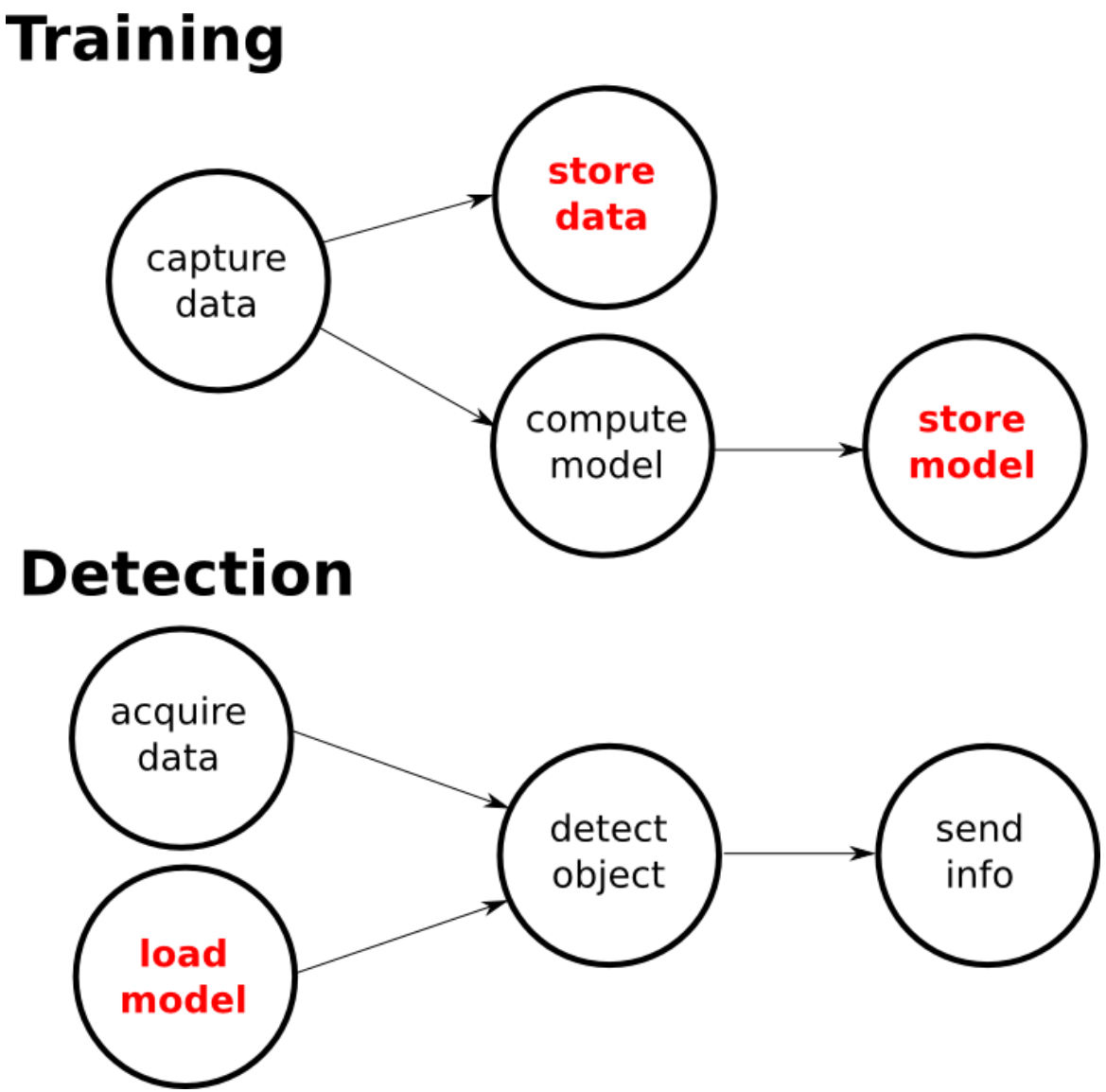
A Step Back (1/5)



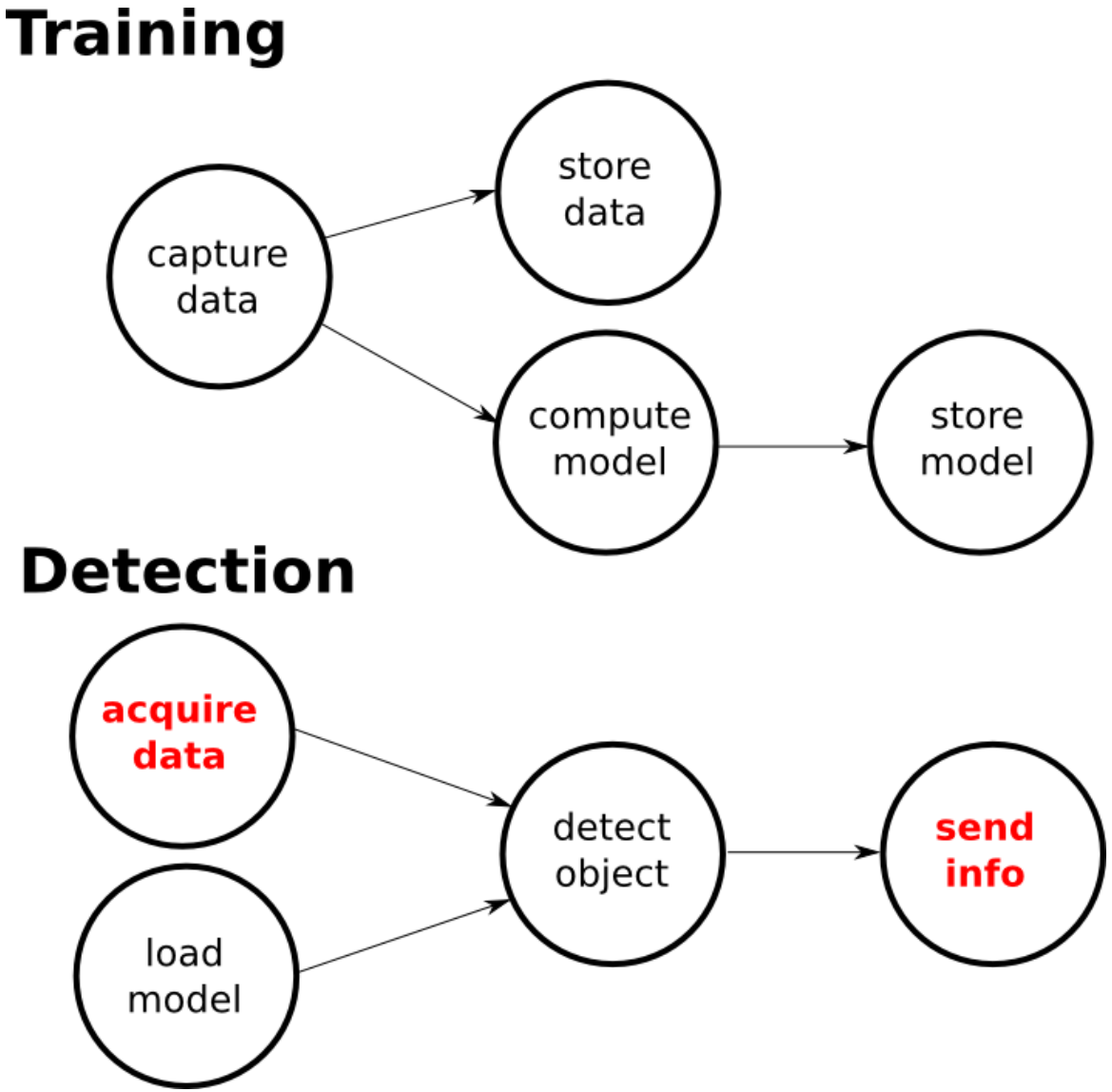
A Step Back (2/5)



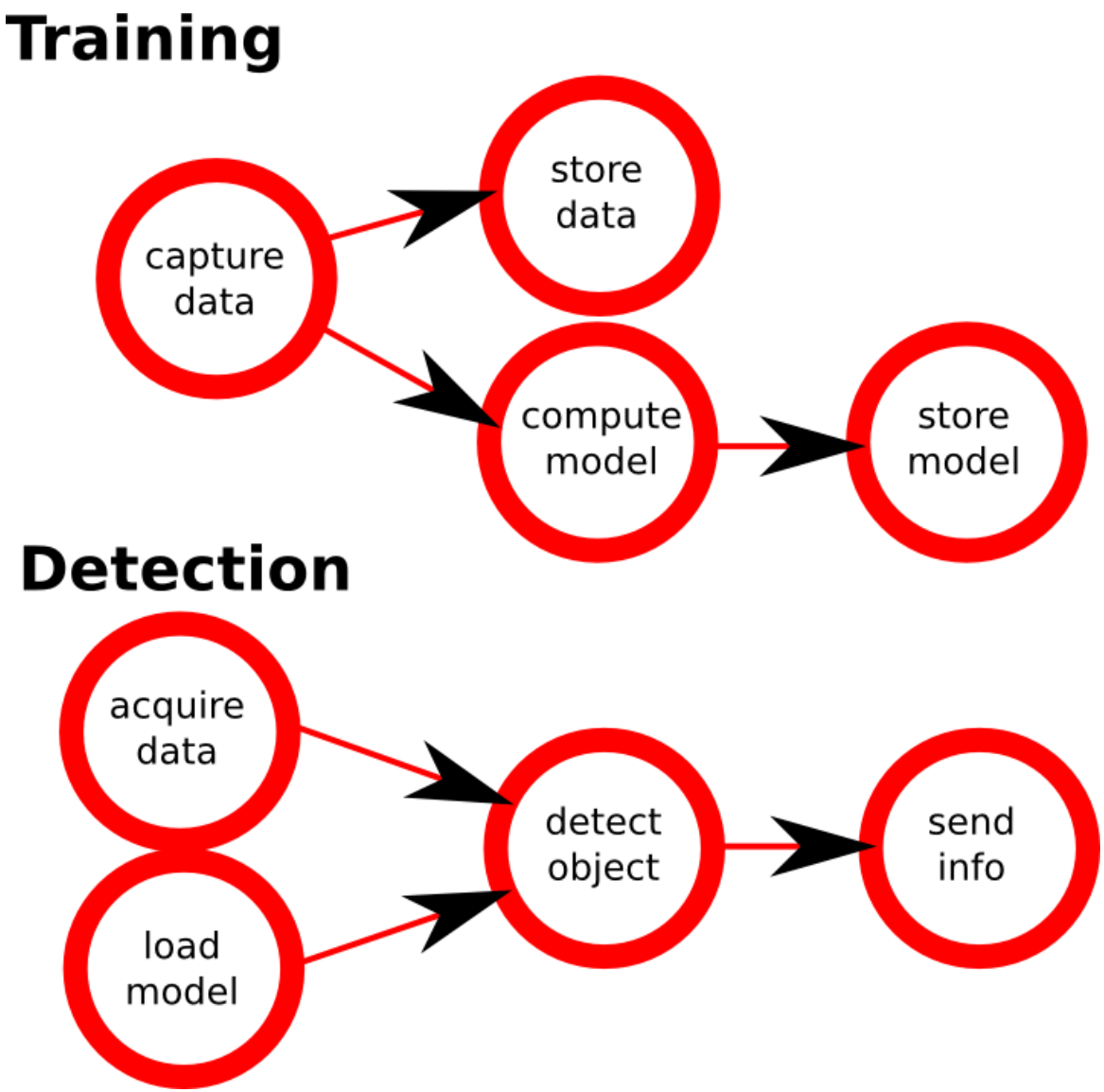
A Step Back (3/5)



A Step Back (4/5)



A Step Back (5/5)



Design Criteria (1/2)

- easy for users: minimal knowledge, flexible
- easy for developers: minimal work to include a new DB/method/IO/pre-post processing
- non-intrusive
- robot/setup/ROS agnostic
- scalable in methods, number of objects, infrastructure

Design Criteria (2/2)

Solutions ?

- reuse as much as possible !
- define some standards
- define a glue framework

Outline

1. Problematic
Previous Work
A Step Back

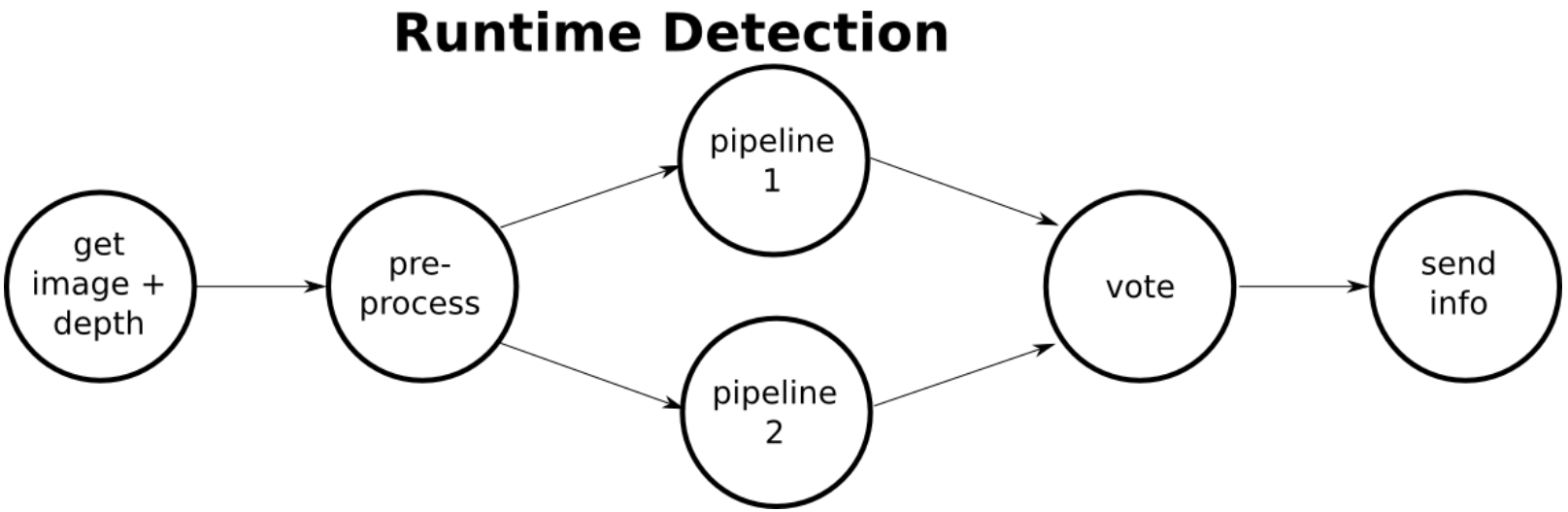
2. A Solution
Ecto
Database
Pipelines
Misc

3. Future Work
Better Pipelines
More Pipelines
More Integration

4. Conclusion

Ecto (1/6)

Motivation:



Ecto (2/6)

Solution: nodelets !

Pros:

- pointer passing
- external access to what is passed

Cons:

- they suck
- no scheduler: no guarantee that the graph actually finishes
- C++ only
- cells are all connected via bonds: mess
- configured through xml: hard to create graphs dynamically

Ecto (3/6)

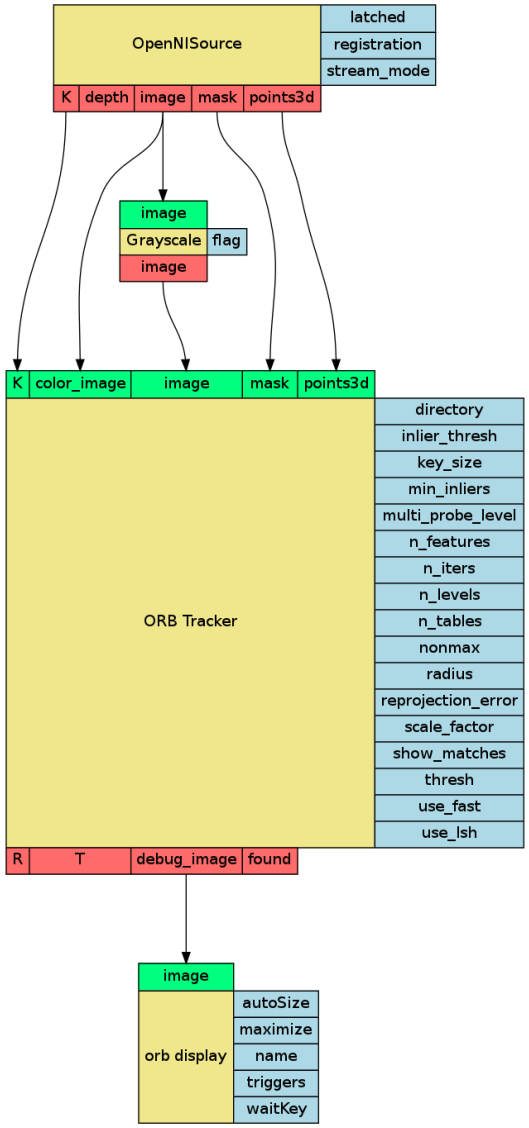
Ecto

Ecto is a hybrid C++/Python framework for organizing computations as directed acyclic graphs.

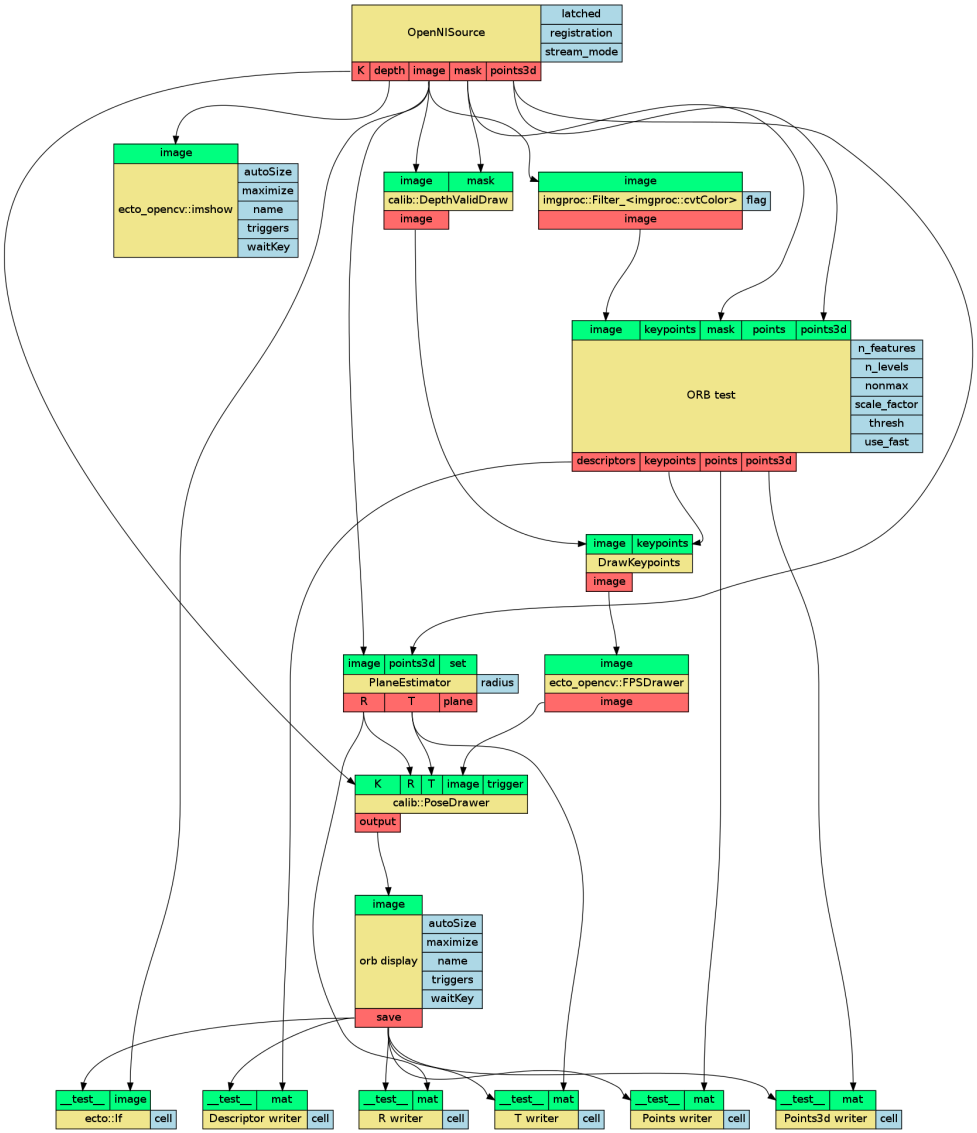
`http://www.ros.org/wiki/ecto`

Main Dev: Troy Straszheim, Ethan Rublee
Extra Dev: Vincent Rabaud, Michael Ferguson, Kurt Konolige

Ecto (4/6)


OpenNISource
latched
registration
stream_mode
K
depth
image
mask
points3d
image
Grayscale
flag
image
K
color_image
image
mask
points3d
ORB Tracker
directory
inlier_thresh
key_size
min_inliers
multi_probe_level
n_features
n_iters
n_levels
n_tables
nonmax
radius
reprojection_error
scale_factor
show_matches
thresh
use_fast
use_lsh
R
T
debug_image
found
image
orb display
autoSize
maximize
name
triggers
waitKey

Ecto (5/6)


OpenNISource
latched
registration
stream_mode
K
depth
image
mask
points3d
image
ecto_opencv::imshow
autoSize
maximize
name
triggers
waitKey
image
mask
calib::DepthValidDraw
image
image
imgproc::Filter_<imgproc::cvtColor>
flag
image
image
keypoints
mask
points
points3d
ORB test
n_features
n_levels
nonmax
scale_factor
thresh
use_fast
descriptors
keypoints
points
points3d
image
keypoints
DrawKeypoints
image
image
points3d
set
PlaneEstimator
radius
R
T
plane
image
ecto_opencv::FPSDrawer
image
K
R
T
image
trigger
calib::PoseDrawer
output
image
orb display
autoSize
maximize
name
triggers
waitKey
save
__test__
image
ecto::If
cell
__test__
mat
Descriptor writer
cell
__test__
mat
R writer
cell
__test__
mat
T writer
cell
__test__
mat
Points writer
cell
__test__
mat
Points3d writer
cell

Ecto (6/6)

Features:

- self-contained and compiled pieces of code: static introspection
- automatic optimization of the graph (e.g. threading)
- code in Python or C++, link in Python or C++ BUT run at C++ speed
- runtime parameter change
- automatic documentation
- automatic benchmarking
- cells can contain anything, even cells
- ROS agnostic, but bindings for ROS, OpenCV, OpenNI ...
- more tutorials/samples/test than code itself

DB (1/3)

Design choices:

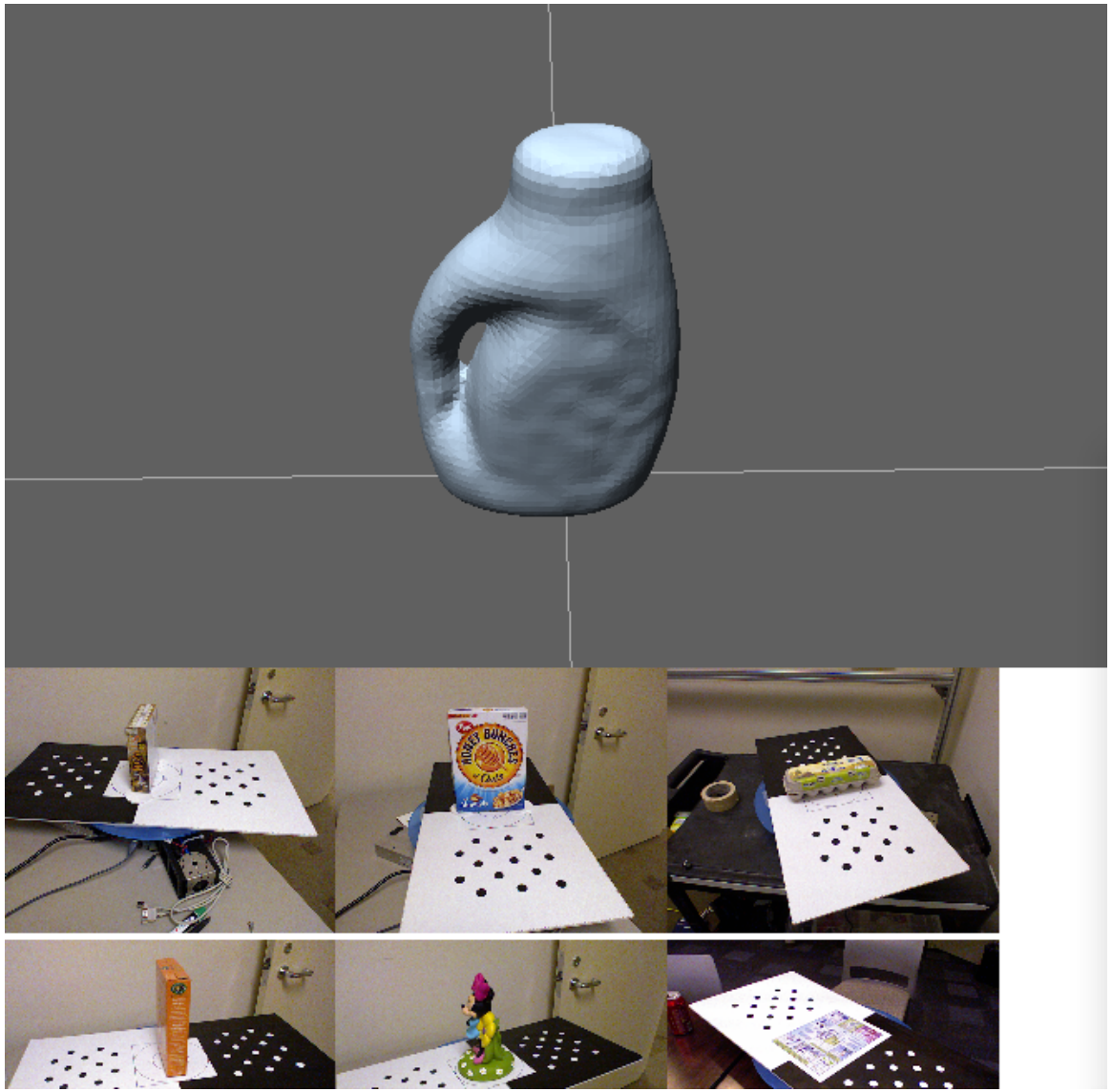
- everything is stored in the DB (capture data, models, object metadata . . .)
- a document is a JSON string with binary blobs
- DB infrastructure agnostic: local/remote, several, diversity (CouchDB, filesystem, household DB)

DB (2/3)

Technically:

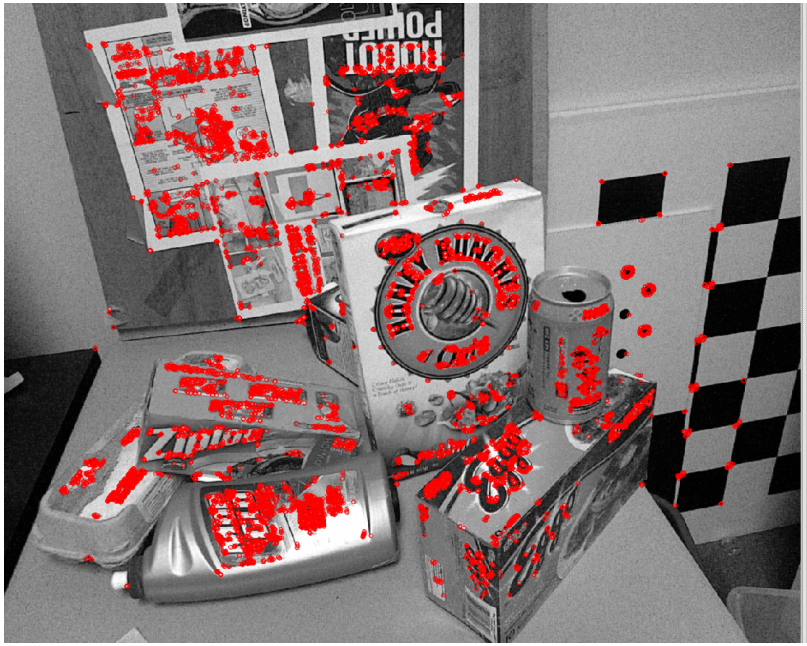
- base class with several simple interactions (get, delete, find . . .)
- a document is a JSON string with binary blobs
- DB infrastructure agnostic: local/remote, several, diversity (CouchDB, filesystem, household DB)

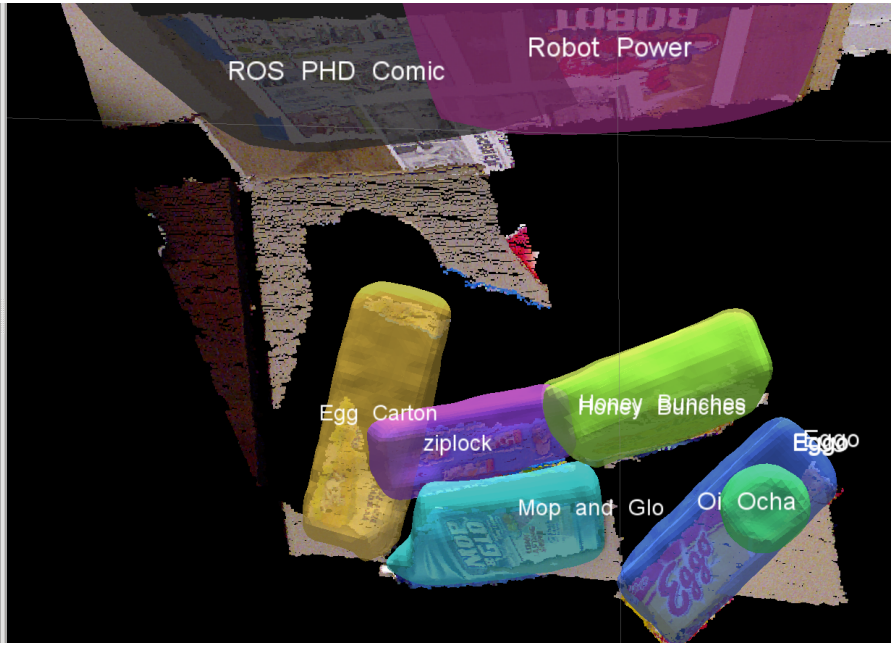
DB (3/3)



Pipelines (1/4)

Textured Object Detection:

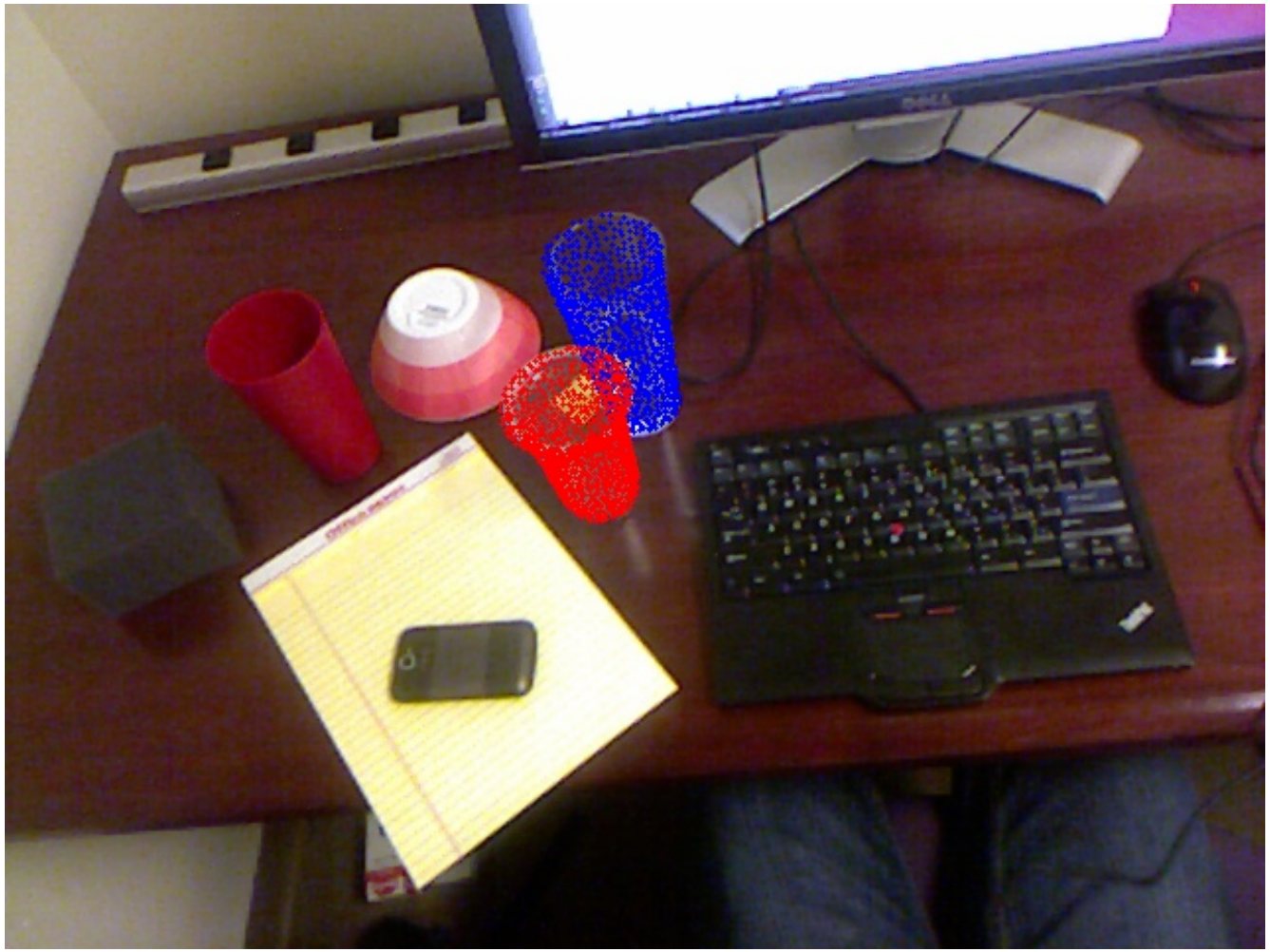




Main Dev: Vincent Rabaud, Ethan Rublee

Pipelines (2/4)

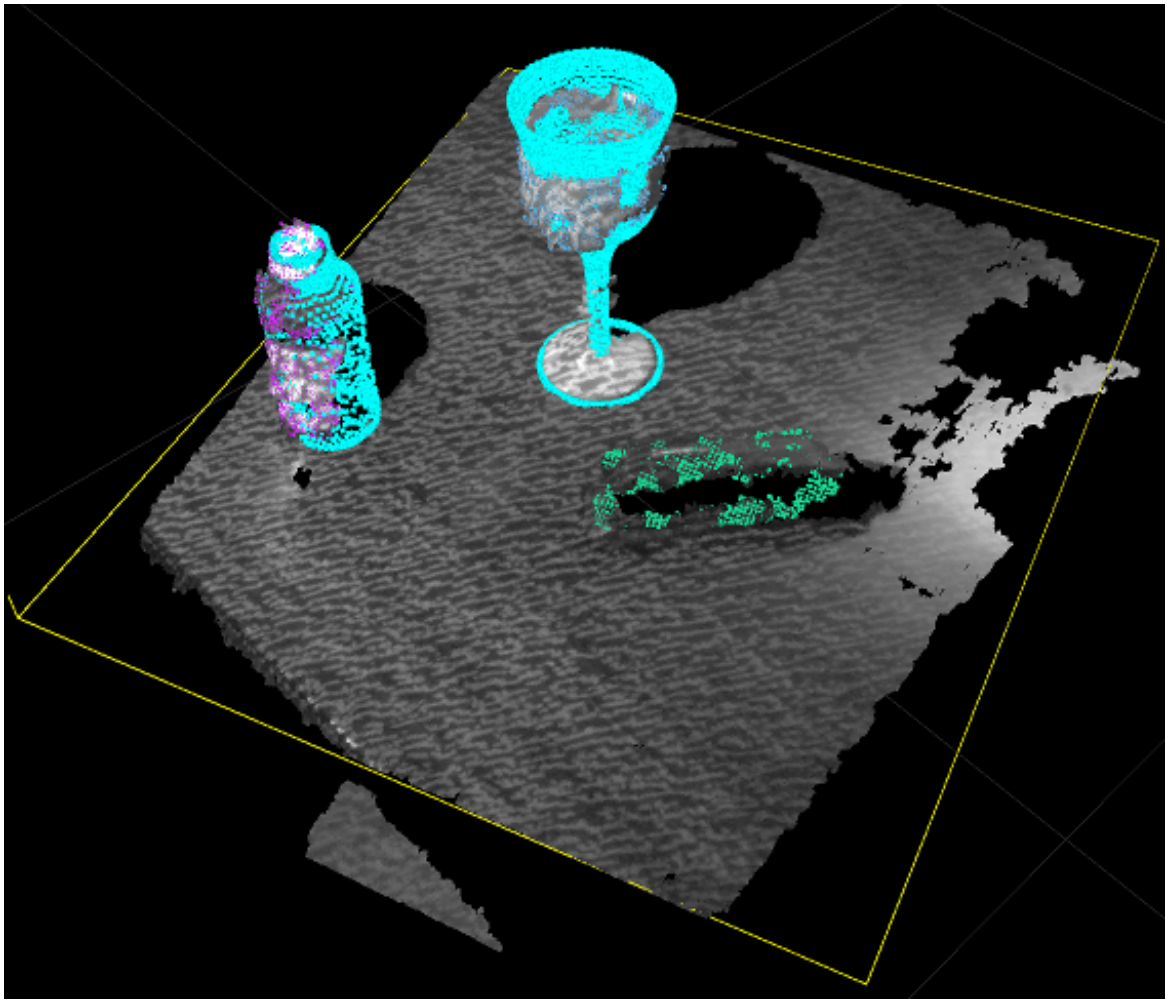
Transparent Objects (cost of integration: 200 lines of code):



Main Dev: Ilya Lysenkov (ICRA 2013)

Pipelines (3/4)

Tabletop:



Main Dev: Marius Muja

Pipelines (4/4)

LINEMOD:

Main Dev: Stefan Hinterstoisser

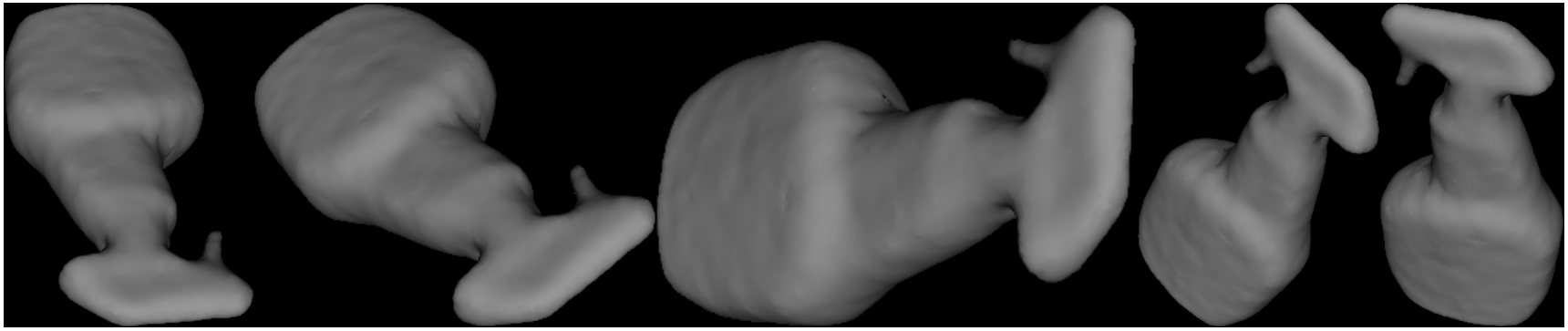
Misc (1/3)

Model Capture:

Main Dev: Ethan Rublee

Misc (2/3)

View Generator:



Main Dev: Vincent Rabaud

Misc (3/3)

ROS integration:

- ROS inputs: topics/ bags
- ROS msg/publisher
- actionlib server
- RViz plugin
- MoveIt! integration

Outline

1. Problematic
Previous Work
A Step Back

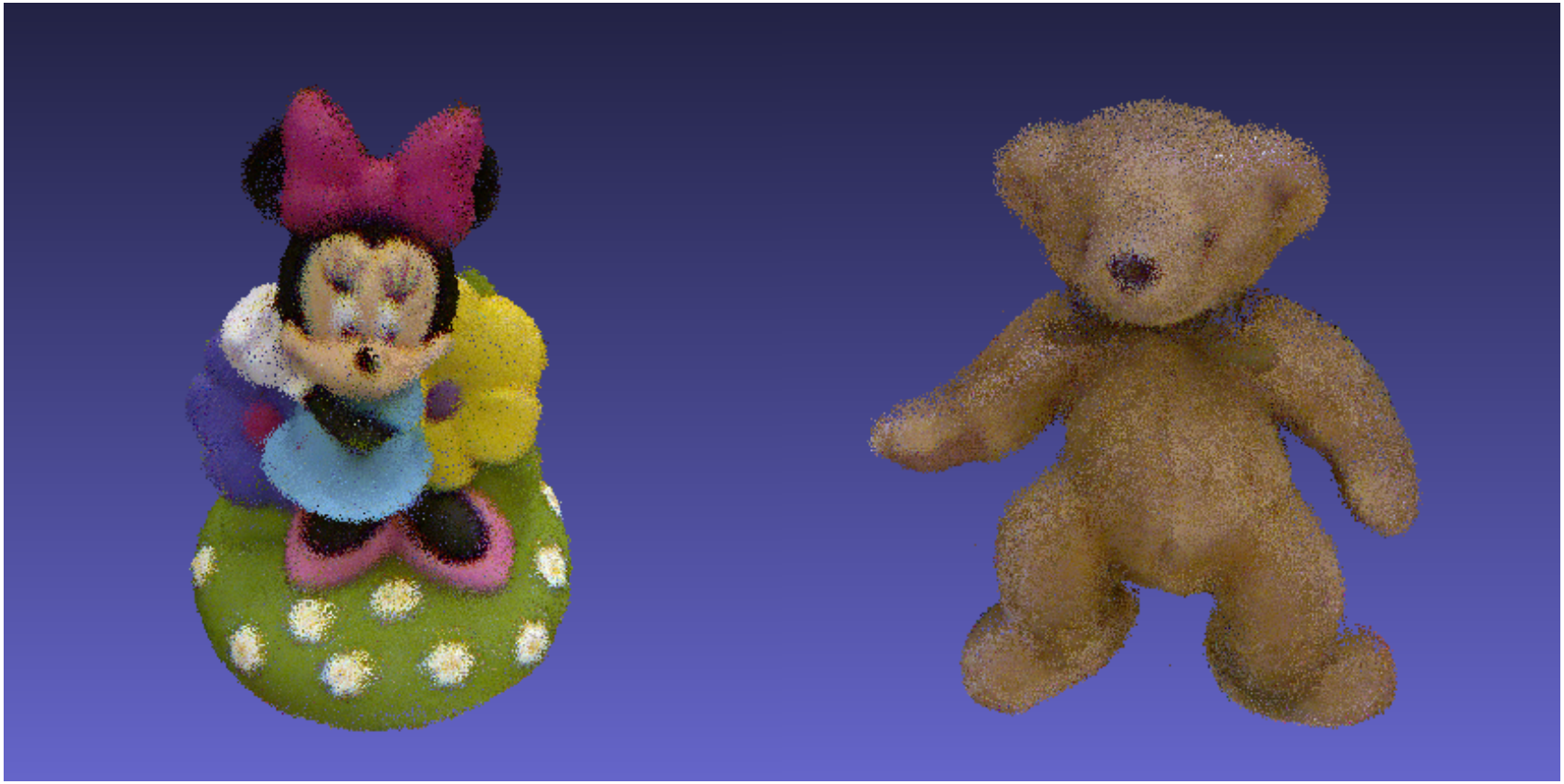
2. A Solution
Ecto
Database
Pipelines
Misc

3. Future Work
Better Pipelines
More Pipelines
More Integration

4. Conclusion

Better Pipelines (1/3)

Better capture:



Main Dev: Maria Dimashova, Ilya Lysenkov (ICRA 2013)

Better Pipelines (2/3)

LINEMOD:

- add a color modality
- add an ICP step

Main Dev: Manuele Tamburrano

Better Pipelines (3/3)

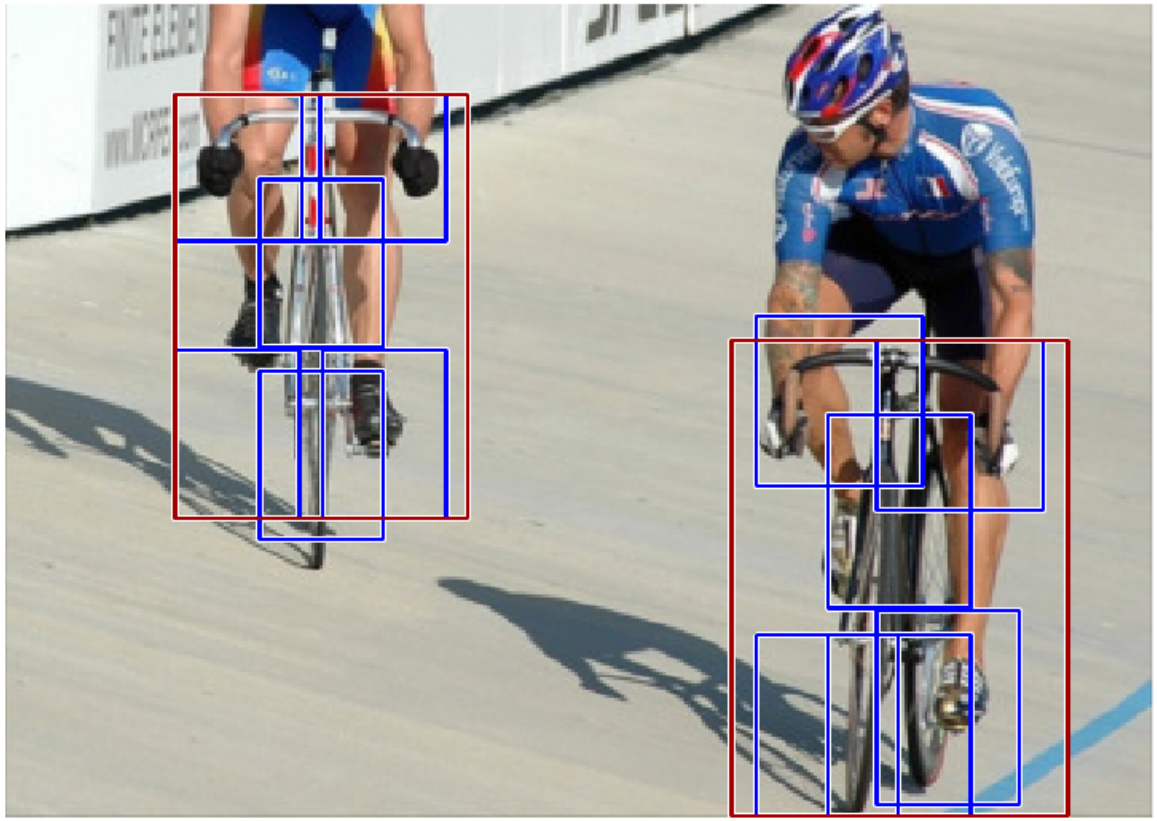
Transparent Objects:

- training without a painted object

Main Dev: Ilya Lysenkov

More Pipelines

Parts based (from Deva Ramanan):



Main Dev: Hilton Bristow

More Integration (1/2)

Object Manipulation:

- still uses the old tabletop
- needs to switch to the DB interface
- ongoing work with MoveIt! switch

More Integration (2/2)

ROS:

- REP for object recognition msgs
- split the DB part in its own package

Outline

1. Problematic
Previous Work
A Step Back

2. A Solution
Ecto
Database
Pipelines
Misc

3. Future Work
Better Pipelines
More Pipelines
More Integration

4. Conclusion

Conclusion (1/2)

BSD packages for Fuerte/Groovy/Hydro:

- ecto
- ecto bindings: ecto-opencv, ecto-pcl, ecto-image-pipeline, ecto-ros, ecto-openni
- core: object-recognition-core, object-recognition-msgs, object-recognition-ros
- pipelines: object-recognition-linemod, object-recognition-tabletop, object-recognition-tod, object-recognition-transparent-objects
- extra pipelines: object-recognition-capture, object-recognition-renderer

Join! (2/2)

Join !

http://ros.org/wiki/object_recognition

http://ros.org/wiki/ecto

Subliminal messages:

- if you've ever used a Nao, please talk to me !
- Aldebaran is hiring heavily, please talk to me !