Fachhochschule Nordwestschweiz
Hochschule für Architektur, Bau und Geomatik

Development of an Acquisition Software for our Image-Based Indoor Mobile Mapping System based on the Robot Operating System (ROS)

GeoPython 2017

Stefan Blaser
Institute of Geomatics Engineering
May 10, 2017

Introduction

Mobile Mapping

- Image-based Mobile Mapping
- Spin-off company iNovitas
- “Google Street View”-like service
- Measurement functionality
- Measurement accuracy:
  - relative: ca. 1 *cm*
  - absolute: ca. $3 - 5$ *cm* (GNSS accuracy)

(Burkhard et al., 2012)

Figure: Mobile Mapping vehicle of FHNW

Introduction

CTI project “BIMAGE”

- Main Goal:
  - Bring outdoor technology to indoor environment

Introduction

CTI project “BIMAGE”

- Main Goal:
  - Bring outdoor technology to indoor environment
- Deliverables:
  - Indoor Mobile Mapping System (IMMS) (absence of GNSS)

Introduction

CTI project “BIMAGE”

- Main Goal:
  - Bring outdoor technology to indoor environment
- Deliverables:
  - Indoor Mobile Mapping System (IMMS) (absence of GNSS)
  - Software to follow changes using mobile devices

Introduction

CTI project “BIMAGE”

- Main Goal:
  - Bring outdoor technology to indoor environment
- Deliverables:
  - Indoor Mobile Mapping System (IMMS) (absence of GNSS)
  - Software to follow changes using mobile devices
  - New navigation concepts for cloud application

Introduction

CTI project “BIMAGE”

- Main Goal:
  - Bring outdoor technology to indoor environment
- Deliverables:
  - Indoor Mobile Mapping System (IMMS) (absence of GNSS)
  - Software to follow changes using mobile devices
  - New navigation concepts for cloud application
  - Sophisticated photogrammetric image post-processing approaches

Fachhochschule Nordwestschweiz
Hochschule für Architektur, Bau und Geomatik

iNovitas

Unterstützt von der KTI

F&E-Projektförderung

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Kommission für Technologie und Innovation KTI

Project team

FHNW IVGI


Prof. Dr. Stephan Nebiker


Stefan Cavegn


Stefan Blaser


Daniel Rettenmund


Robert Wüest

iNovitas AG


Dr. Hannes Eugster

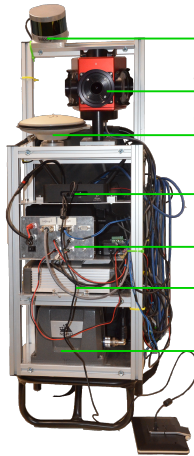

Benjamin Loesch

KTI

Alain Tanner

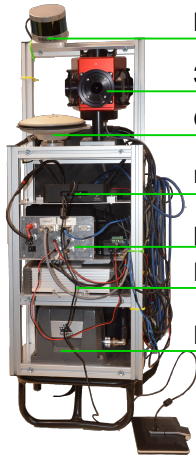
Dr. Rolf Wohlgemuth

Indoor mobile mapping system prototype



- multi sensor system
- weight: 30 *kg* (66.1 *lbs*)
- research prototype
- hardware components might change during project time

Indoor mobile mapping system prototype

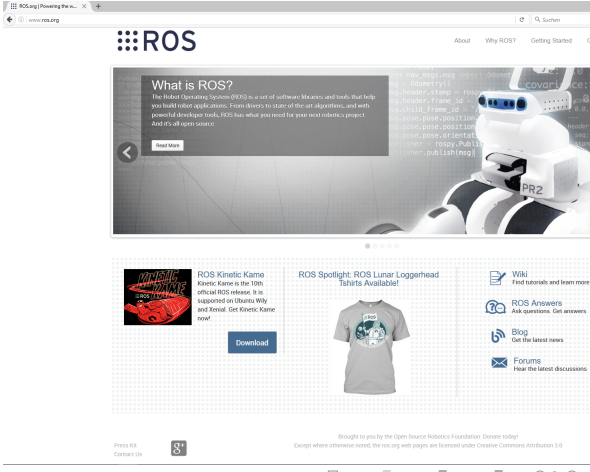


- multi sensor system
- weight: 30 *kg* (66.1 *lbs*)
- research prototype
- hardware components might change during project time
- acquisition software requirements:
  - Modular software architecture
  - Free open source software (FOSS)
  - Existing modules for hardware support

Robot Operating System (ROS)

- Software framework for robots
- Free and Open-Source
- Graph-based communication layer
- Collection of tools and libraries
- Multi-lingual
  - C++
  - **Python**

  Quigley et al. (2009)
- `www.ros.org`



Graph-based communication layer

Node software module, process

Parameter state of a specific node

Messages data in a predefined structure

Topic where messages go through

Service allows synchronous communication (request / response)

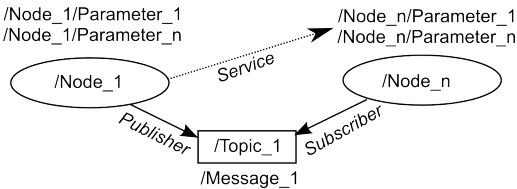
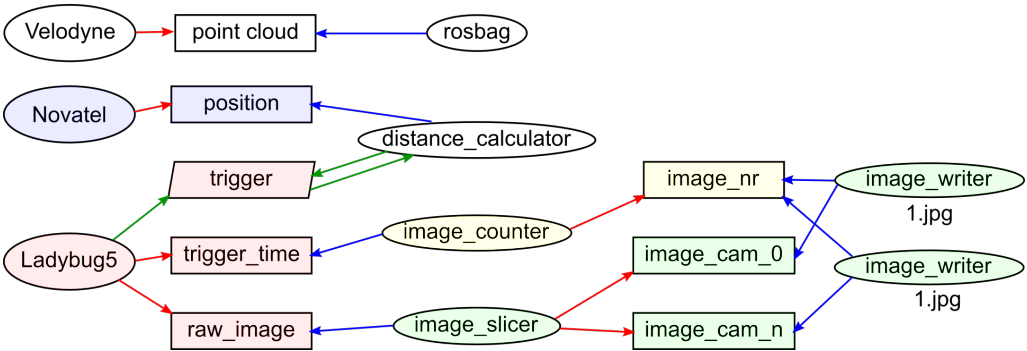
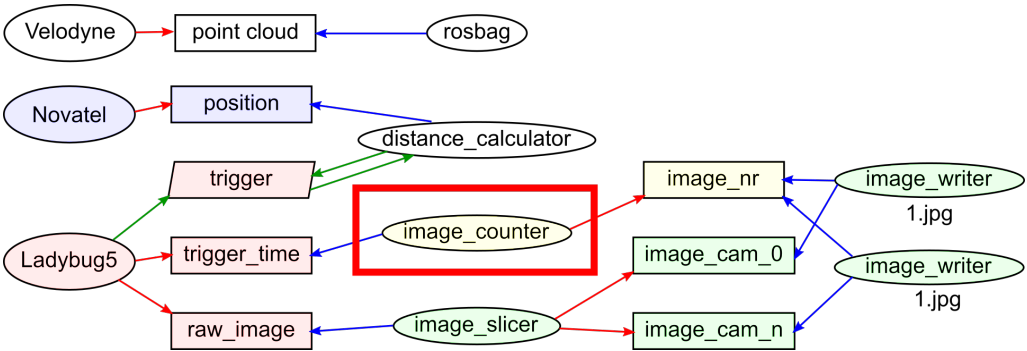


Figure: ROS communication principle

Acquisition software design



Acquisition software design



ImageEventCounter – node structure

```
#!/usr/bin/env python
import rospy
from std_msgs.msg import *

class ImageEventCounter():
    [...]

def main():
    [...]

if __name__ == '__main__':
    main()
```

ImageEventCounter – main function

```
def main():
    rospy.init_node("ImageEventCounter",
                    anonymous=True)
    image_event_counter_node = ImageEventCounter()
    try:
        rospy.spin()
    except KeyboardInterrupt:
        pass
```

ImageEventCounter – read out parameter

```
class ImageEventCounter():
    def __init__(self):
        # read out parameter
        self.counter = \
            rospy.get_param('counter_start', 0)
        subscriber_tn = \
            rospy.get_param('subscriber_name',
                            '/trigger_time')
        publisher_tn = \
            rospy.get_param('publisher_name',
                            '/image_nr')
        [...]
```

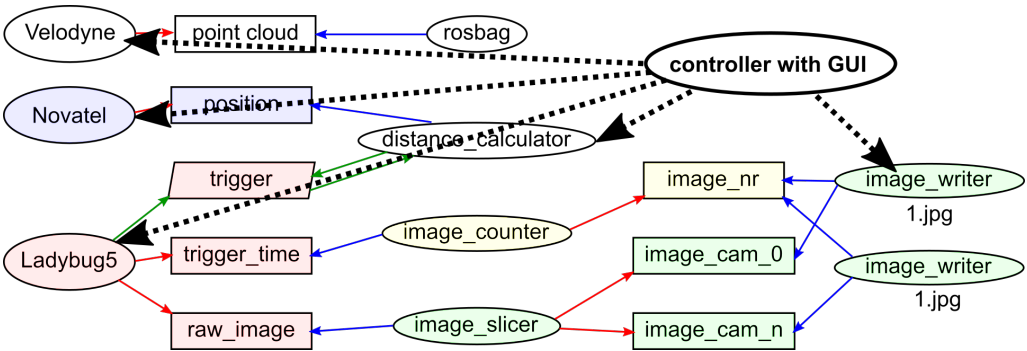
ImageEventCounter – publisher / subscriber

```
class ImageEventCounter():
    def __init__(self):
        [...]
        self.trigger_time_sub = \
            rospy.Subscriber(subscriber_tn,
                             Time,
                             self.subscriber_callback)
        self.trigger_count_pub = \
            rospy.Publisher(publisher_tn,
                            UInt32,
                            queue_size=1)
    def subscriber_callback(self, time):
        [...]
```

ImageEventCounter – subscriber callback

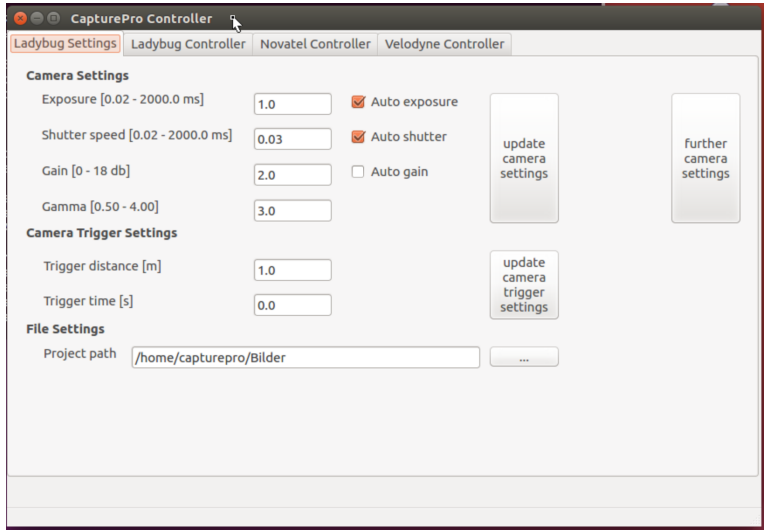
```
class ImageEventCounter():
    def __init__(self):
        [...]
    def subscriber_callback(self, time):
        self.counter += 1
        self.trigger_count_pub(UInt32(\
                                self.counter))
```

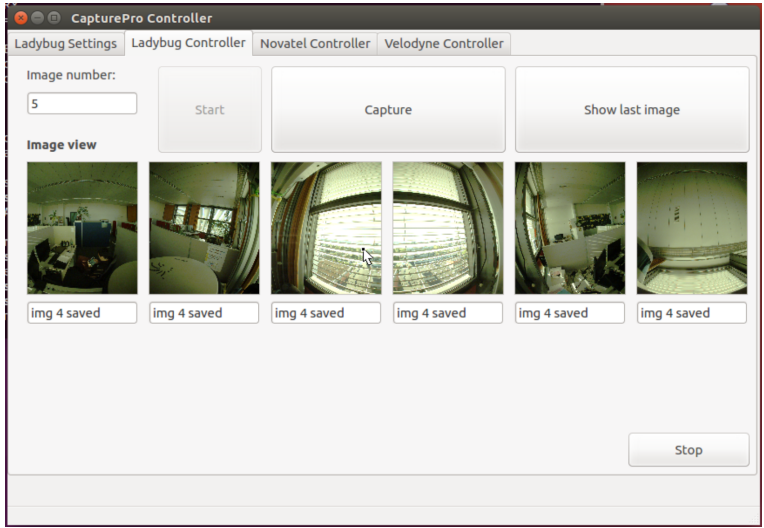
Controller node

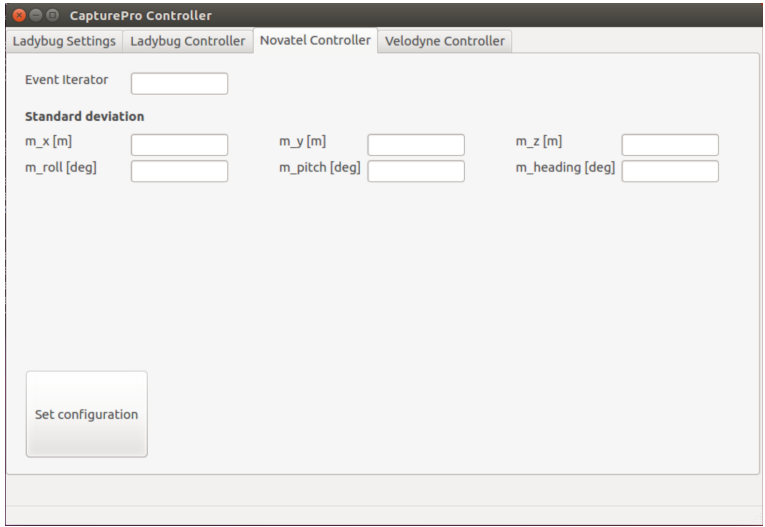


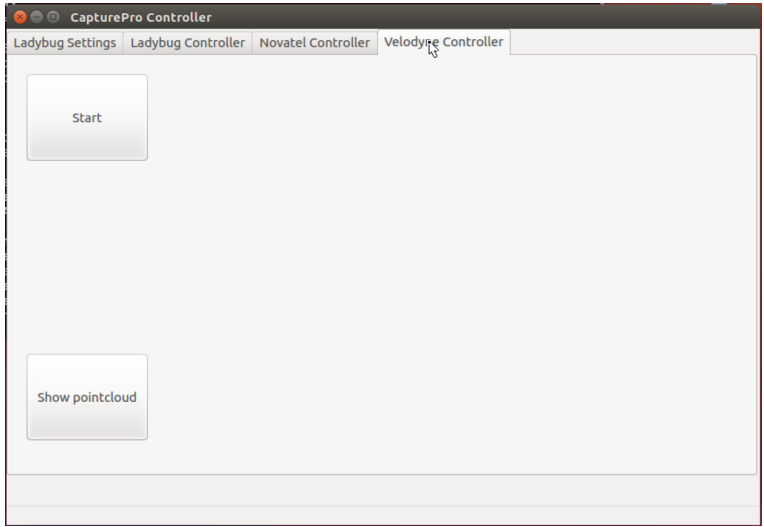
Controller node

- dynamic parameter changes
- change node settings
- launches and kills nodes
- GUI
  - developed with pyqt4
  - put ROS node into pyqt4-loop
  - *rospy.spin()* becomes obivious


CapturePro Controller
Ladybug Settings
Ladybug Controller
Novatel Controller
Velodyne Controller
Camera Settings
Exposure [0.02 - 2000.0 ms]
1.0
Auto exposure
Shutter speed [0.02 - 2000.0 ms]
0.03
Auto shutter
Gain [0 - 18 db]
2.0
Auto gain
Gamma [0.50 - 4.00]
3.0
update camera settings
further camera settings
Camera Trigger Settings
Trigger distance [m]
1.0
Trigger time [s]
0.0
update camera trigger settings
File Settings
Project path
/home/capturepro/Bilder
...


CapturePro Controller
Ladybug Settings
Ladybug Controller
Novatel Controller
Velodyne Controller
Image number:
5
Start
Capture
Show last image
Image view
img 4 saved
img 4 saved
img 4 saved
img 4 saved
img 4 saved
img 4 saved
Stop


CapturePro Controller
Ladybug Settings
Ladybug Controller
Novatel Controller
Velodyne Controller
Event Iterator
Standard deviation
m_x [m]
m_y [m]
m_z [m]
m_roll [deg]
m_pitch [deg]
m_heading [deg]
Set configuration

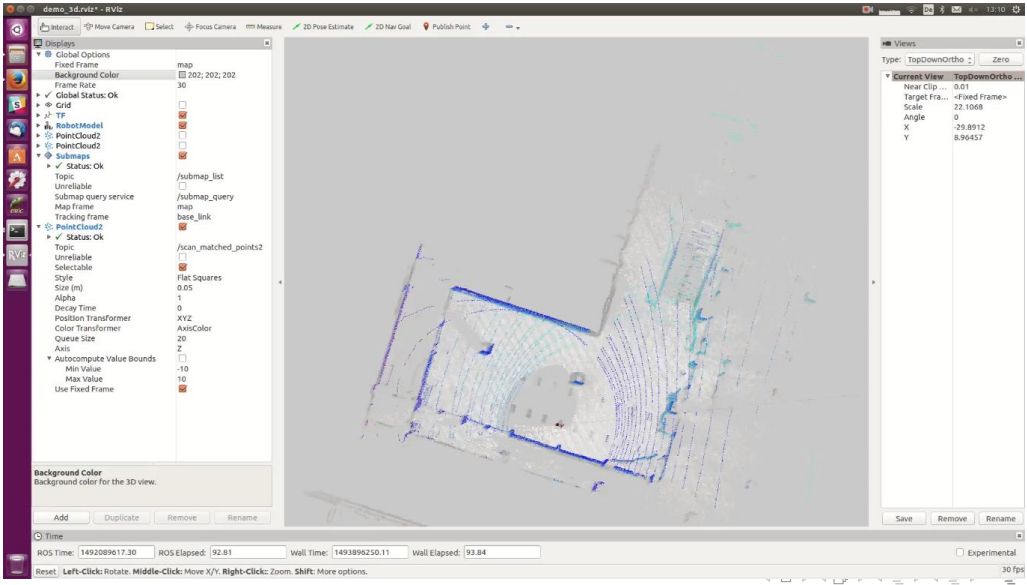

CapturePro Controller
Ladybug Settings
Ladybug Controller
Novatel Controller
Velodyne Controller
Start
Show pointcloud

SLAM (post processing)

- simultaneous localization and mapping (SLAM)
- different SLAM techniques (lidar-based, visual-based, ...)
- some SLAM algorithms are supporting loop closure
- possible subtitution of GNSS in indoor environment

SLAM (post processing)

- simultaneous localization and mapping (SLAM)
- different SLAM techniques (lidar-based, visual-based, ...)
- some SLAM algorithms are supporting loop closure
- possible subtitution of GNSS in indoor environment
- Google cartographer
  - lidar-based SLAM, lidar and IMU data fused within a kalman filter
  - Open source software since 05.10.2016
  - ROS support
  - both, 2D and 3D SLAM version
  - loop closure (Hess et al., 2016)


demo_3d.rviz* - RViz
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame
map
Background Color
202; 202; 202
Frame Rate
30
Global Status: Ok
Grid
TF
RobotModel
PointCloud2
PointCloud2
Submaps
Status: Ok
Topic
/submap_list
Unreliable
Submap query service
/submap_query
Map frame
map
Tracking frame
base_link
PointCloud2
Status: Ok
Topic
/scan_matched_points2
Unreliable
Selectable
Style
Flat Squares
Size (m)
0.05
Alpha
1
Decay Time
0
Position Transformer
XYZ
Color Transformer
AxisColor
Queue Size
20
Axis
Z
Autocompute Value Bounds
Min Value
-10
Max Value
10
Use Fixed Frame
Background Color
Background color for the 3D view.
Add
Duplicate
Remove
Rename
Views
Type: TopDownOrtho
Zero
Current View
TopDownOrtho ...
Near Clip ...
0.01
Target Fra...
<Fixed Frame>
Scale
22.1068
Angle
0
X
-29.8912
Y
8.96457
Save
Remove
Rename
Time
ROS Time:
1492089617.30
ROS Elapsed:
92.81
Wall Time:
1493896250.11
Wall Elapsed:
93.84
Experimental
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click: Zoom. Shift: More options.
30 fps

Conclusion

- IMMS acquisition software successfully implemented → based on ROS
- graph-based architecture → flexibility

Conclusion

- IMMS acquisition software successfully implemented → based on ROS
- graph-based architecture → flexibility
- numerous existing free open source ROS packages available
- from hardware node up to SLAM algorithm

Conclusion

- IMMS acquisition software successfully implemented → based on ROS
- graph-based architecture → flexibility
- numerous existing free open source ROS packages available
- from hardware node up to SLAM algorithm
- wide ROS community → quasi-standard in robotics

Conclusion

- IMMS acquisition software successfully implemented → based on ROS
- graph-based architecture → flexibility
- numerous existing free open source ROS packages available
- from hardware node up to SLAM algorithm
- wide ROS community → quasi-standard in robotics
- learning curve

Outlook

- Indoor localization $\rightarrow$ Replacement of INS by SLAM
- real time SLAM
- weight loss about 10 *kg*
- use SLAM map as progress control
- replacement of ladybug5 camera by a stereo panorama camera
- several GUI optimizations
- ...

Literature

Burkhard, J., Cavegn, S., Barmettler, A. and Nebiker, S. (2012). Stereovision Mobile Mapping : System Design and Performance Evaluation, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XXXIX-B5* **XXXIX**(September): 453–458.
URL: *http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B5/453/2012/isprsarchives-XXXIX-B5-453-2012.html*

Hess, W., Kohler, D., Rapp, H. and Andor, D. (2016). Real-time loop closure in 2D LIDAR SLAM, *Proceedings - IEEE International Conference on Robotics and Automation*, Vol. 2016-June, pp. 1271–1278.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R. and Mg, A. (2009). ROS: an open-source Robot Operating System, *ICRA* **3**: 5.
URL: *http://pub1.willowgarage.com/ konolige/cs225B/docs/quigley-icra2009-ros.pdf*

Thank you for your attention!

- Questions?

Stefan Blaser
Research Associate
Institute of Geomatics Engineering
stefan.blaser@fhnw.ch
Twitter: @stibe88