

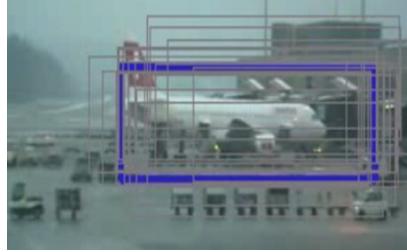
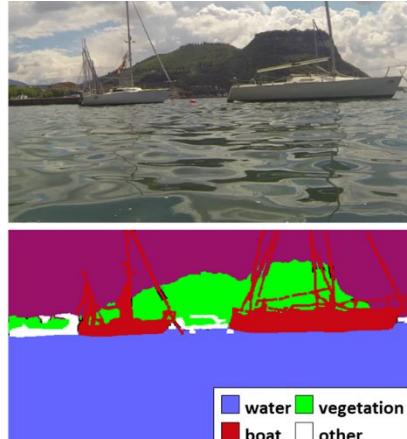
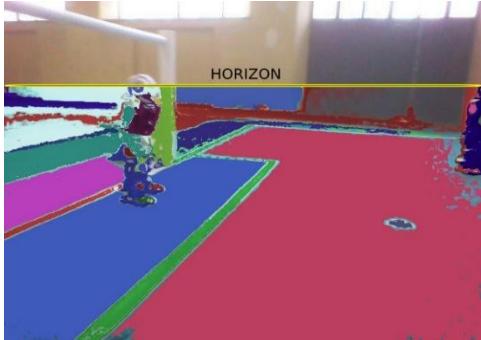


UNIVERSITÀ DEGLI STUDI DELLA BASILICATA

Corso di Sistemi Informativi
A.A. 2018/19

rosbag

Maggio 2019



References and credits

Queste slide si basano sul materiale contenuto nel libro

YoonSeok Pyo, HanCheol Cho, RyuWoon Jung, TaeHoon Lim,
“*ROS Robot Programming - A Handbook Written by TurtleBot3 Developers*”
<http://www.robotis.com/service/download.php?no=719>

rosbag

- I dati contenuti nei messaggi ROS possono essere registrati in appositi file
- Il file che contiene i messaggi prende il nome di **bag** e ha l'estensione “.bag”
- Il vantaggio offerto dai file di bag è quello di avere una registrazione che può essere utilizzata **più volte**, riproducendo ogni volta l'esatto scenario operativo in cui la bag è stata registrata

rosbag per i dati dei sensori

- Un esempio dell'utilità dei file di bag è dato dalla registrazione dei messaggi contenenti i dati prodotti dai sensori del robot
- Durante gli esperimenti con il robot reale, i dati dei sensori possono essere registrati in una bag
- I messaggi registrati possono essere poi caricati senza la necessità di ripetere l'esperimento, permettendo così di sviluppare con maggiore facilità algoritmi che richiedano modifiche frequenti dei parametri

Usare rosbag

rosbag è un package ROS per creare, riprodurre e comprimere bag di messaggi. Una bag è un file contenente i dati relativi a messaggi serializzati

- `rosbag record` □ record all the topics
- `rosbag info bag-name` □ info on the recorded bag
- `rosbag play --pause bag-name` □ play the recorded bag, starting paused
- `rosbag play -r #number bag-name` □ play the recorded bag at rate #number

Comandi rosbag

Command	Description
rosbag record [OPTION] [TOPIC_NAME]	Record the message of a specific topic on the bsg file
rosbag info [FILE_NAME]	Check information of a bag file
rosbag play [FILE_NAME]	Play a specific bag file
rosbag compress [FILE_NAME]	Compress a specific bag file
rosbag decompress [FILE_NAME]	Decompresses a specific bag file
rosbag filter [INPUT_FILE] [OUTPUT_FILE] [OPTION]	Create a new bag file with the specific content removed
rosbag reindex bag [FILE_NAME]	Reindex
rosbag check bag [FILE_NAME]	Check if the specific bag file can be played in the current system
rosbag fix [INPUT_FILE] [OUTPUT_FILE] [OPTION]	Fix the bag file version that was saved as an incompatible version

Esempio rosbag

Apriamo un terminal e digitiamo
\$ roscore

```
roscore http://localhost:11311/
bloisi@bloisi-U36SG:~$ roscore
... logging to /home/bloisi/.ros/log/4d85da46-576b-11e8-9e4c-2709ac87ed01/roslaunch-bloisi-U36SG-2511.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://localhost:38804/
ros_comm version 1.12.13

SUMMARY
=====

PARAMETERS
* /rosdistro: kinetic
* /rosversion: 1.12.13

NODES

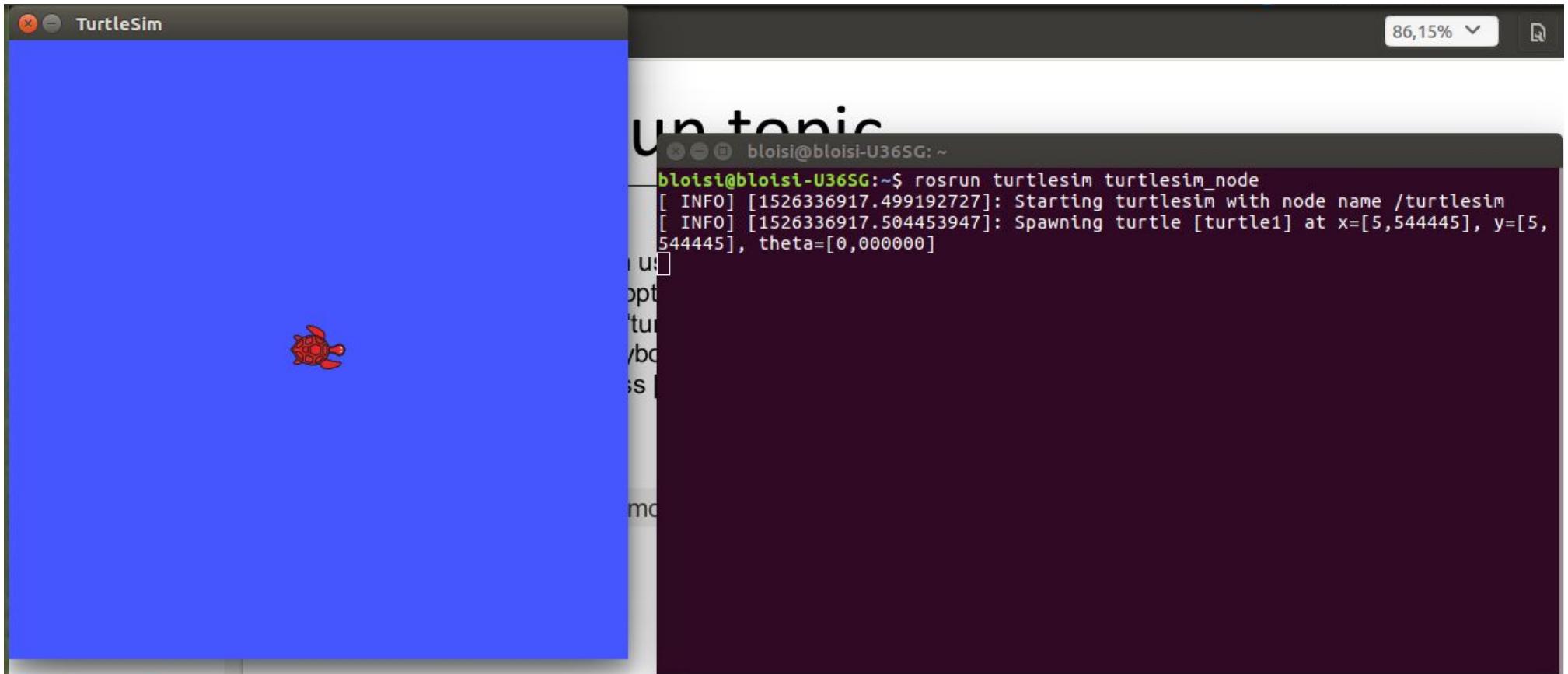
auto-starting new master
process[master]: started with pid [2523]
ROS_MASTER_URI=http://localhost:11311/

setting /run_id to 4d85da46-576b-11e8-9e4c-2709ac87ed01
process[rosout-1]: started with pid [2536]
started core service [/rosout]
```

Esempio rosbag – turtlesim

Apriamo un secondo terminal e digitiamo

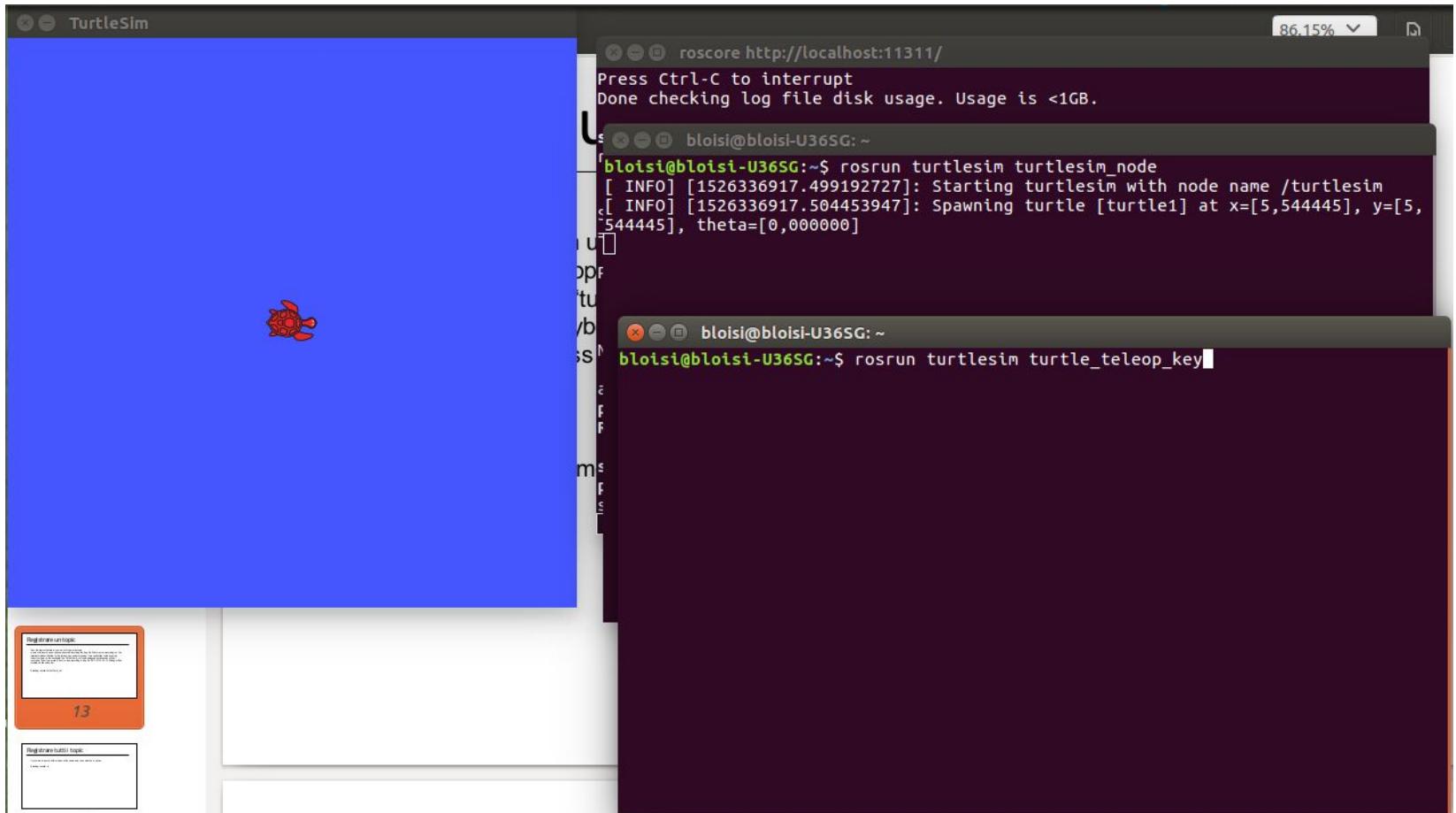
```
$ rosrun turtlesim turtlesim_node
```



Esempio rosbag - teleop

Apriamo un terzo terminal e digitiamo

```
$ rosrun turtlesim turtle_teleop_key
```

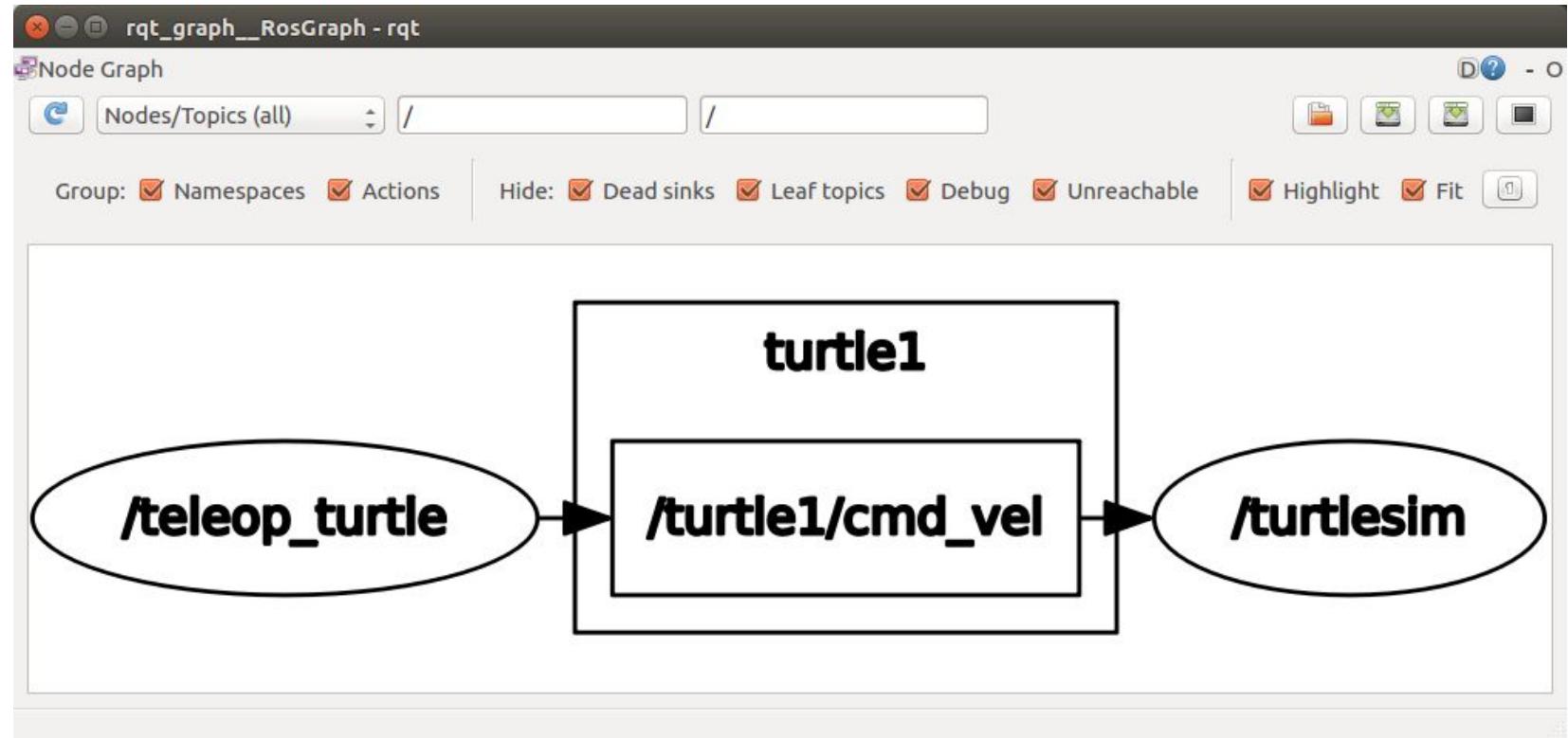


rqt_graph

Apriamo un quarto terminal e digitiamo

```
$ rqt_graph
```

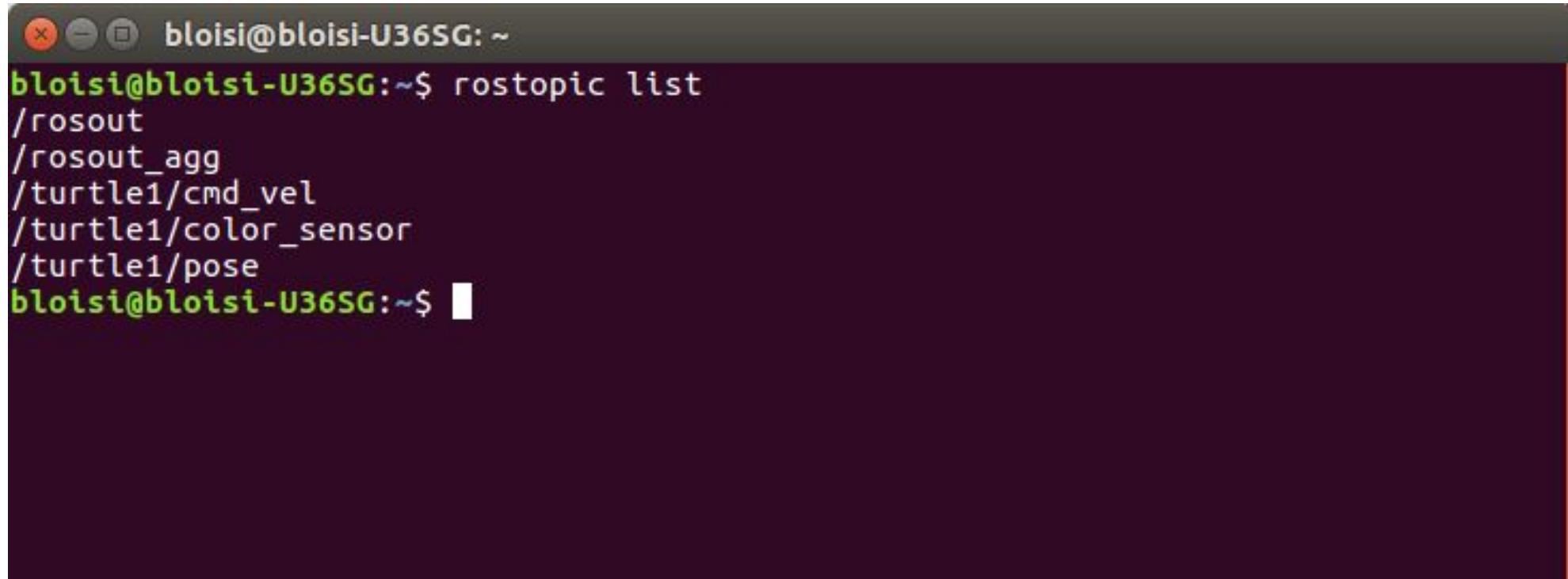
per verificare che i due nodi siano in collegamento tra loro



rostopic

Controlliamo anche la lista dei topic attivi

```
$ rostopic list
```

A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains the command "rostopic list" followed by a list of active topics: /rosout, /rosout_agg, /turtle1/cmd_vel, /turtle1/color_sensor, and /turtle1/pose. The terminal has a dark background with light-colored text and standard window controls at the top.

```
bloisi@bloisi-U36SG: ~$ rostopic list
/rosout
/rosout_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
bloisi@bloisi-U36SG: ~$
```

Registrare un topic

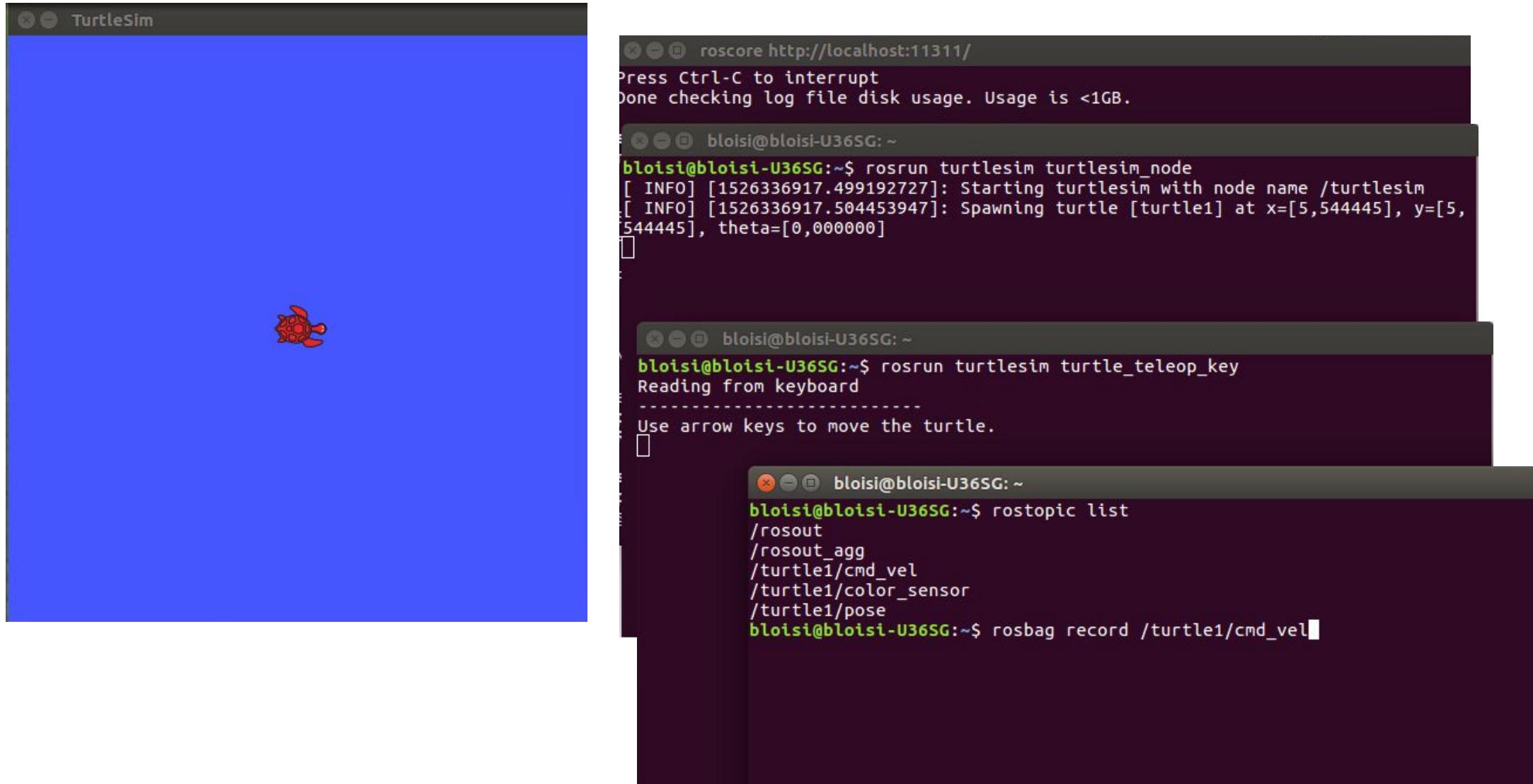
Tra tutti i topic attivi, possiamo scegliere quali registrare usando i comandi e le opzioni di rosbag

```
rosbag record <topic name>
```

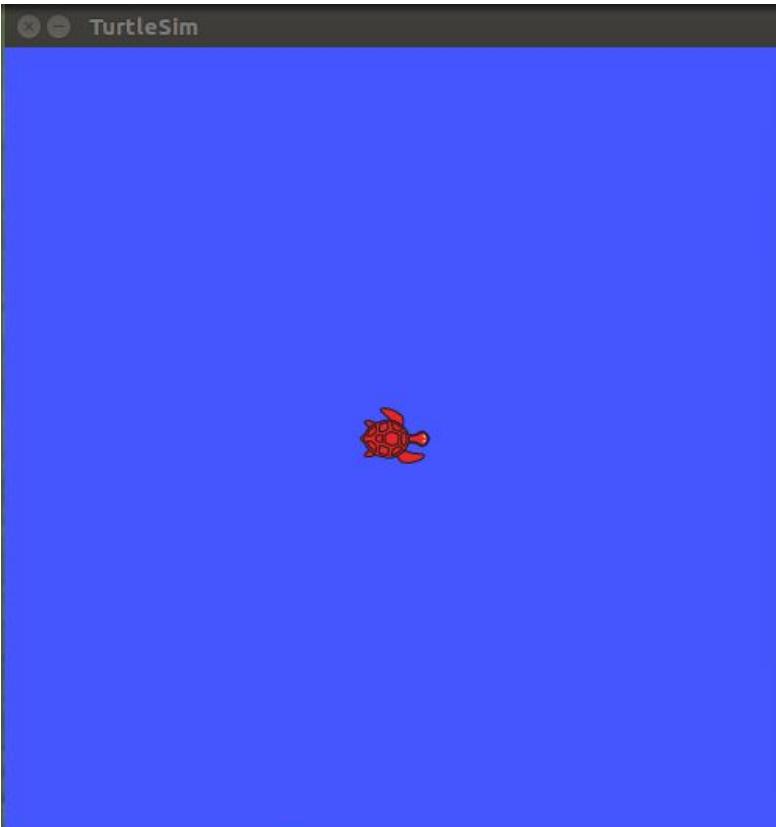
Per esempio, per registrare i comandi inviati tramite cmd_vel digitiamo

```
$ rosbag record /turtle1/cmd_vel
```

Esempio - Registrare un topic



Esempio – Registrazione in corso...



```
roscore http://localhost:11311/
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

bloisi@bloisi-U36SG:~$ rosrun turtlesim turtlesim_node
[ INFO] [1526336917.499192727]: Starting turtlesim with node name /turtlesim
[ INFO] [1526336917.504453947]: Spawning turtle [turtle1] at x=[5,544445], y=[5,
544445], theta=[0,000000]

bloisi@bloisi-U36SG:~$ rosrun turtlesim turtle_teleop_key
Reading from keyboard
-----
Use arrow keys to move the turtle.

bloisi@bloisi-U36SG:~$ rostopic list
/rosout
/rosout_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
bloisi@bloisi-U36SG:~$ rosbag record /turtle1/cmd_vel
[ INFO] [1526337037.406675213]: Subscribing to /turtle1/cmd_vel
[ INFO] [1526337037.414123426]: Recording to 2018-05-15-00-30-37.bag.
```

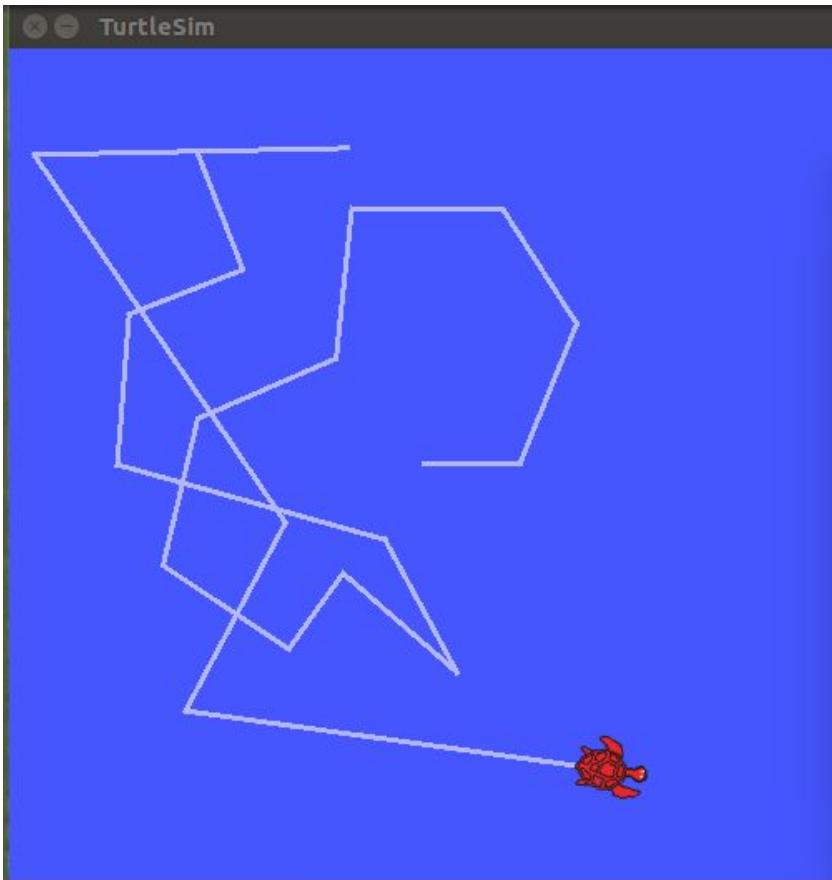
Registrare tutti i topic

Se si vogliono registrare tutti i topic attivi, si può usare l'opzione -a

```
$ rosbag record -a
```

Terminare la registrazione

[Ctrl-C] ci permette di terminare la registrazione della bag



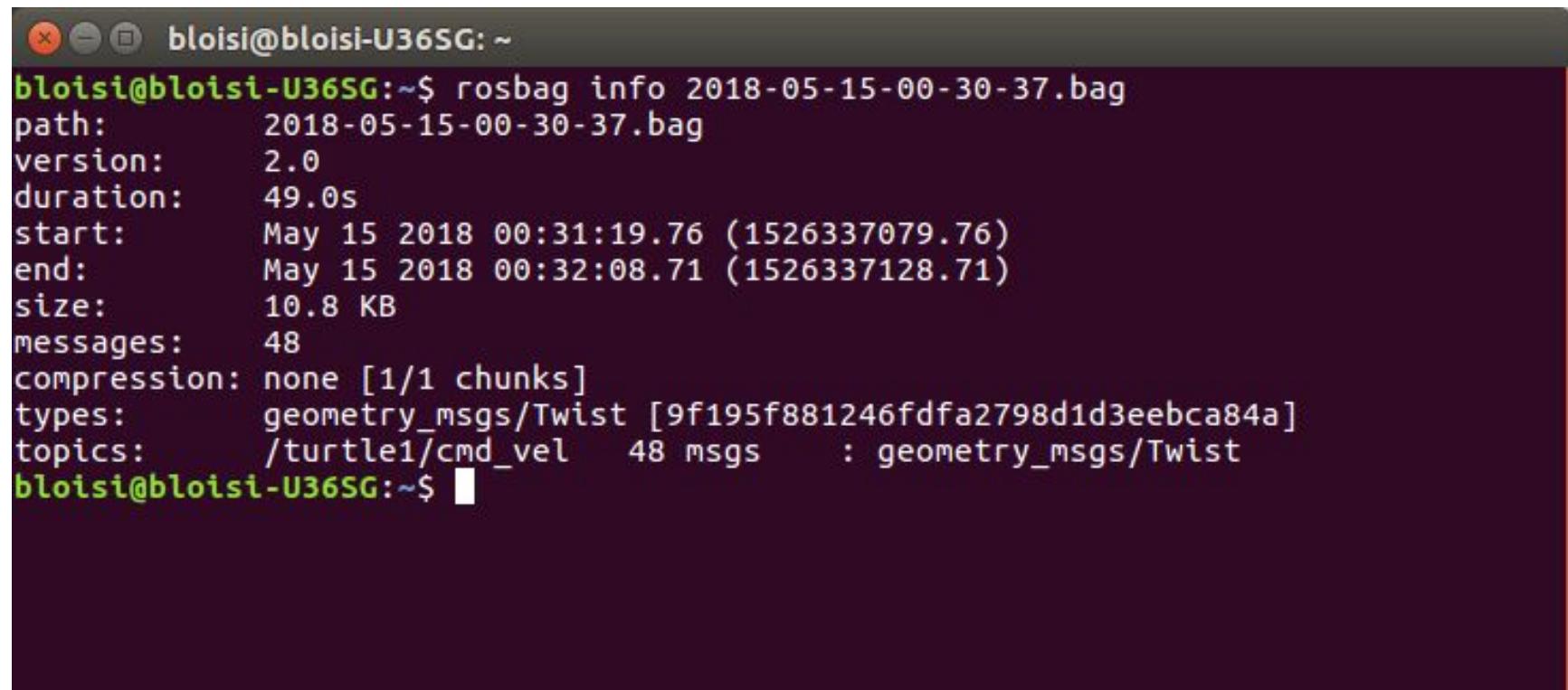
```
bloisi@bloisi-U36SG:~$ rostopic list
/rosout
/rosout_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
bloisi@bloisi-U36SG:~$ rosbag record /turtle1/cmd_vel
[ INFO] [1526337037.406675213]: Subscribing to /turtle1/cmd_vel
[ INFO] [1526337037.414123426]: Recording to 2018-05-15-00-30-37.bag.
^Cbloisi@bloisi-U36SG:~$
```

rosbag info

Il comando `info` stampa a video informazioni sulla bag fornita come parametro.

Nel nostro esempio, avremo informazioni sul file `2018-05-15-00-30-37.bag` digitando

```
$ rosbag info 2018-05-15-00-30-37.bag
```

A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains the command "rosbag info 2018-05-15-00-30-37.bag" and its output. The output provides details about the bag file, including its path, version, duration, start and end times, size, number of messages, compression type, message types, and topics.

```
bloisi@bloisi-U36SG:~$ rosbag info 2018-05-15-00-30-37.bag
path:      2018-05-15-00-30-37.bag
version:   2.0
duration:  49.0s
start:     May 15 2018 00:31:19.76 (1526337079.76)
end:      May 15 2018 00:32:08.71 (1526337128.71)
size:     10.8 KB
messages: 48
compression: none [1/1 chunks]
types:    geometry_msgs/Twist [9f195f881246fdfa2798d1d3eebca84a]
topics:   /turtle1/cmd_vel 48 msgs : geometry_msgs/Twist
bloisi@bloisi-U36SG:~$
```

rosbag play

Avendo a disposizione una bag registrata, si può riprodurla tramite

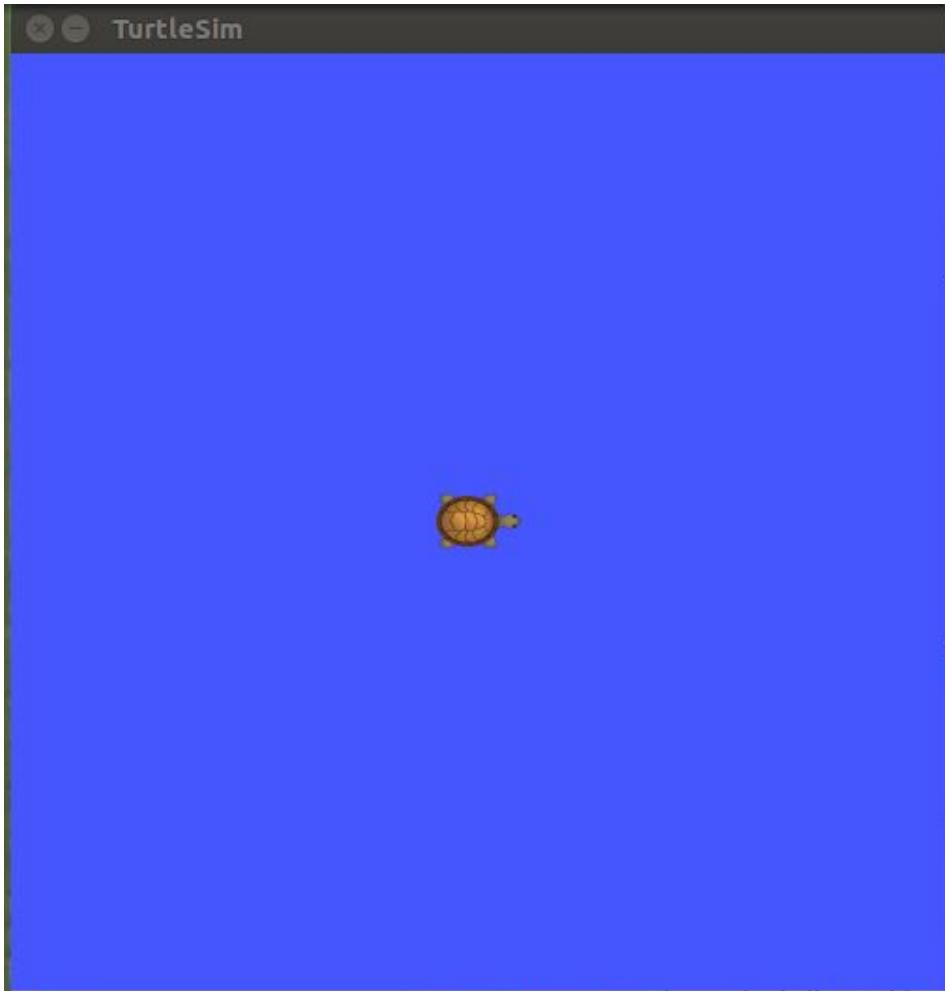
```
rosbag play <bagfile name>
```

Per esempio, per riprodurre la bag

1. Terminiamo tutti i nodi attivi
2. Lanciamo il nodo turtlesim_node
3. Digitiamo

```
$ rosbag play 2018-05-15-00-30-37.bag
```

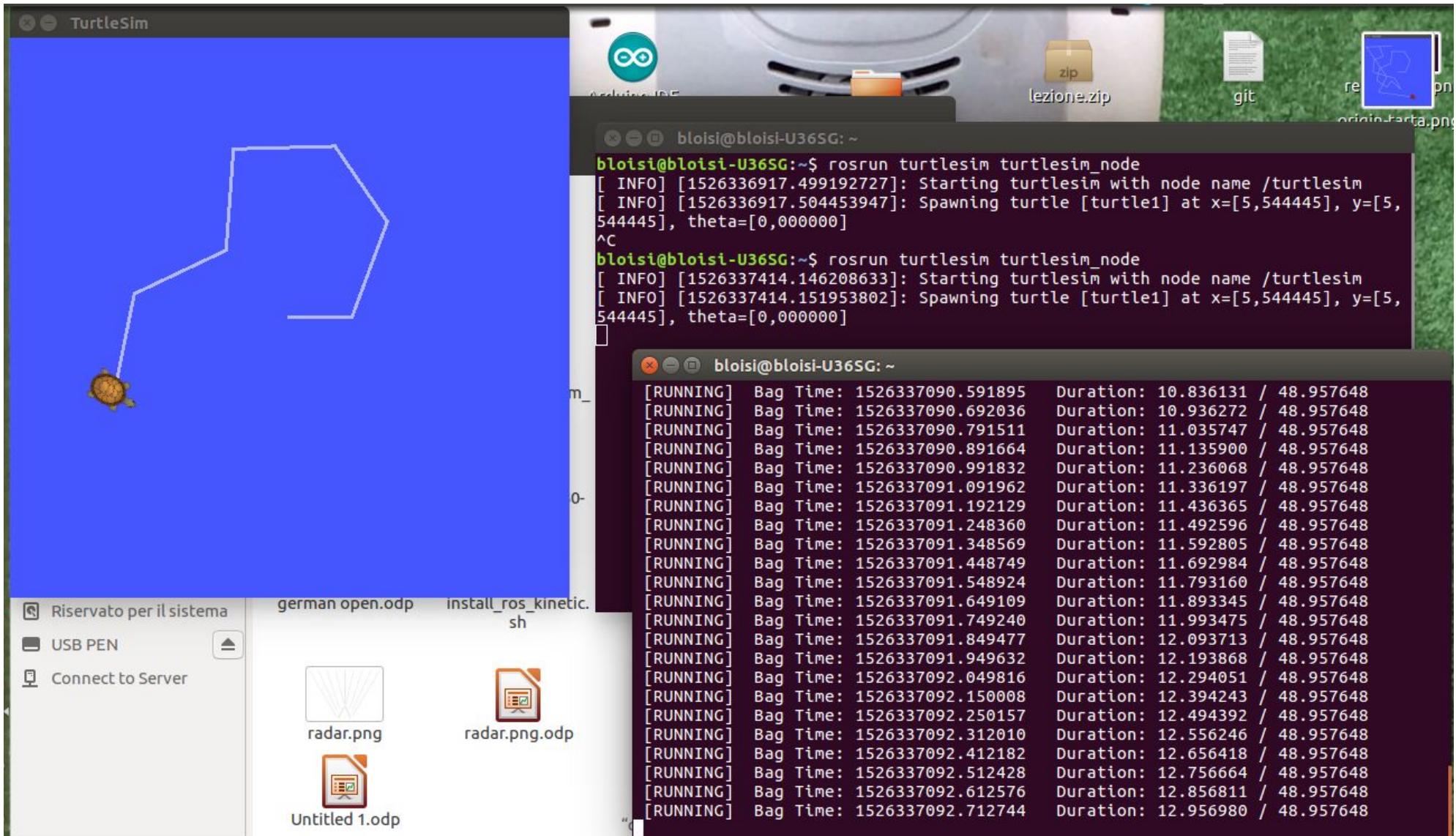
rosbag play – esecuzione



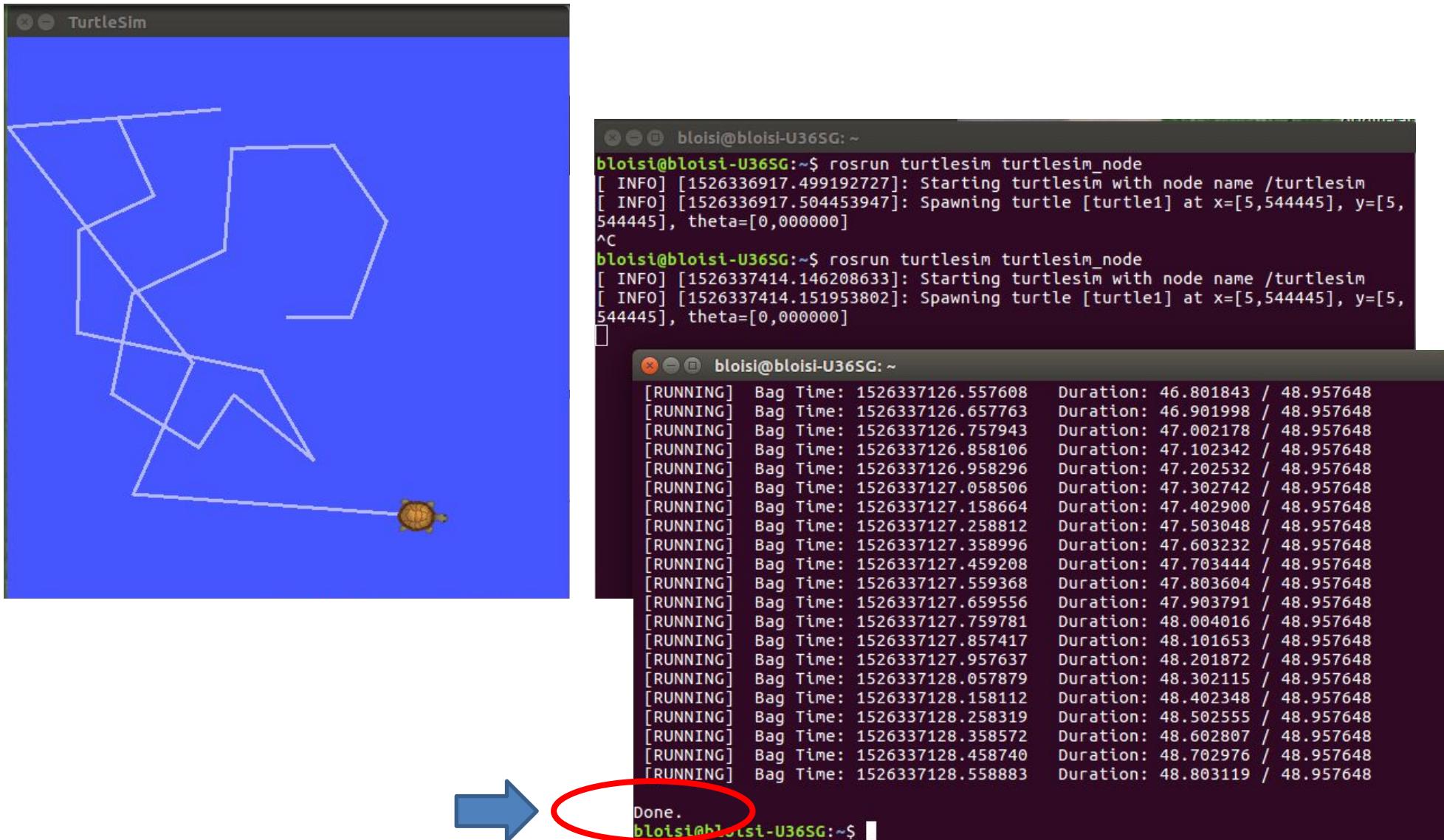
```
bloisi@bloisi-U36SG:~$ rosrun turtlesim turtlesim_node
[ INFO] [1526336917.499192727]: Starting turtlesim with node name /turtlesim
[ INFO] [1526336917.504453947]: Spawning turtle [turtle1] at x=[5,544445], y=[5,544445], theta=[0,000000]
^C
bloisi@bloisi-U36SG:~$ rosrun turtlesim turtlesim_node
[ INFO] [1526337414.146208633]: Starting turtlesim with node name /turtlesim
[ INFO] [1526337414.151953802]: Spawning turtle [turtle1] at x=[5,544445], y=[5,544445], theta=[0,000000]
[ ]
```

```
bloisi@bloisi-U36SG:~$ rosbag play 2018-05-15-00-30-37.bag
```

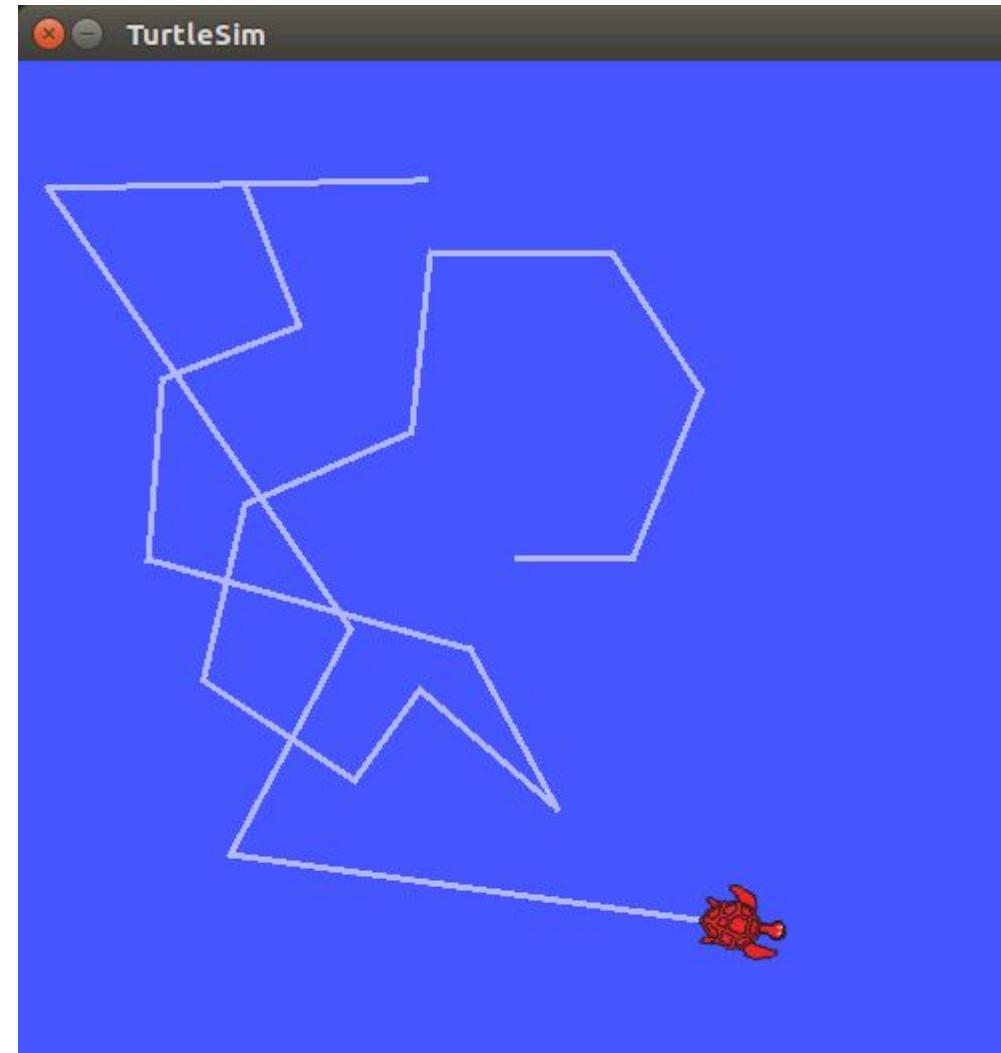
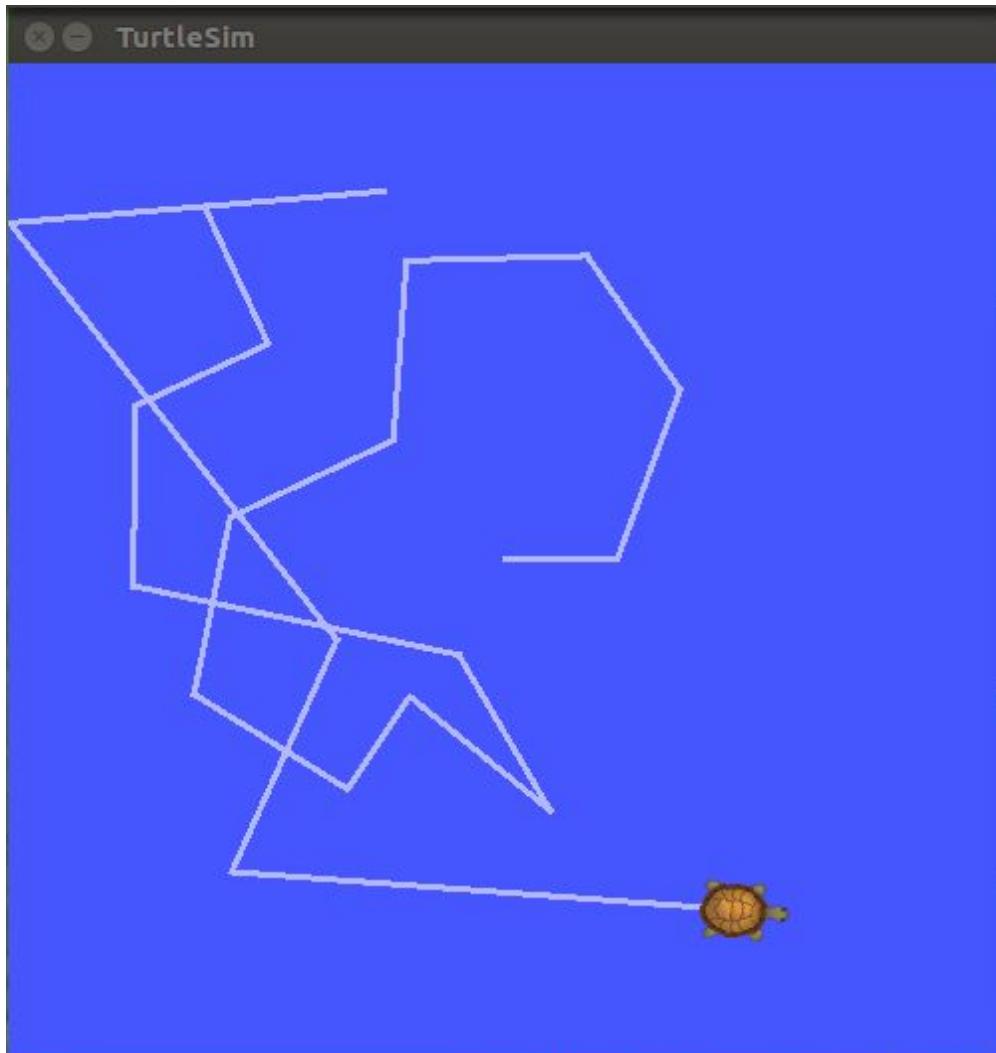
rosbag play – esecuzione in corso



rosbag play – risultato finale



rosbag play – confronto



I bag file possono essere molto grandi

Un bag file registrato per un breve periodo di tempo comporta la creazione di file aventi dimensioni contenute

Se, invece, si ha bisogno di registrare messaggi per un lungo periodo di tempo, allora la dimensione del bag file può crescere fino ad occupare molta memoria

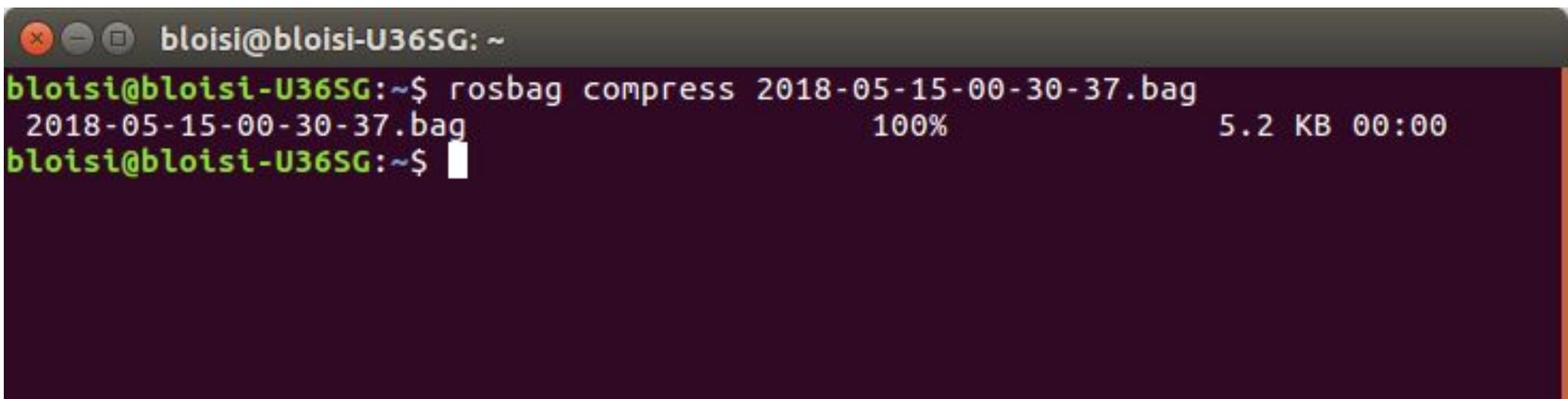
Si provi per esempio a scaricare la ROS bag a questo indirizzo

https://drive.google.com/file/d/1F8pd_Cc5n67cMkWdvTZphpi7zeсRJDEJ/view?usp=sharing

rosbag compress

ROS fornisce la possibilità di comprimere i bag file grazie all'opzione compress

```
$ rosbag compress 2018-05-15-00-30-37.bag
```



The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "bloisi@bloisi-U36SG: ~". The command "rosbag compress 2018-05-15-00-30-37.bag" is entered at the prompt. The output shows the progress: "100%" completion, "5.2 KB" size, and "00:00" duration. The terminal ends with a "\$" prompt.

```
bloisi@bloisi-U36SG:~$ rosbag compress 2018-05-15-00-30-37.bag
2018-05-15-00-30-37.bag          100%      5.2 KB 00:00
bloisi@bloisi-U36SG:~$
```

rosbag compress – esecuzione

```
bloisi@bloisi-U36SG: ~
bloisi@bloisi-U36SG:~$ rosbag compress 2018-05-15-00-30-37.bag
2018-05-15-00-30-37.bag                                100%      5.2 KB 00:00
bloisi@bloisi-U36SG:~$ ls
2018-05-15-00-30-37.bag
2018-05-15-00-30-37.orig.bag
```

rosbag decompress

Per riportare il bag file al suo formato originale, è possibile utilizzare decompress

```
$ rosbag decompress 2018-05-15-00-30-37.bag
```

rosbag con immagini

Una bag può contenere qualunque tipo di dato sia possibile inviare tramite i messaggi ROS

Le bag possono essere molto utili per la registrazione di dati provenienti da telecamere montate su robot

In particolare, essendo presente un timestamp per ogni immagine, è possibile riprodurre fedelmente lo stream dati del sensore usato per effettuare le riprese

Image message

[**sensor_msgs/Image Message**](#)

File: [**sensor_msgs/Image.msg**](#)

Raw Message Definition

```
# This message contains an uncompressed image
# (0, 0) is at top-left corner of image
#
Header header           # Header timestamp should be acquisition time of image
# Header frame_id should be optical frame of camera
# origin of frame should be optical center of camera
# +x should point to the right in the image
# +y should point down in the image
# +z should point into to plane of the image
# If the frame_id here and the frame_id of the CameraInfo
# message associated with the image conflict
# the behavior is undefined
uint32 height            # image height, that is, number of rows
uint32 width              # image width, that is, number of columns
#
# The legal values for encoding are in file src/image_encodings.cpp
# If you want to standardize a new string format, join
# ros-users@lists.sourceforge.net and send an email proposing a new encoding.
string encoding           # Encoding of pixels -- channel meaning, ordering, size
# taken from the list of strings in include/sensor_msgs/image_encodings.h
```

cv_bridge



[About](#) | [Support](#) | [Discussion Forum](#) | [Service Status](#) | [Q&A answers.ros.org](#)

Search:

Documentation

Browse Software

News

Download

cv_bridge

[indigo](#) [kinetic](#) [lunar](#) [melodic](#)

Show EOL distros:

[Documentation Status](#)

[vision_opencv](#): [cv_bridge](#) | [image_geometry](#)

Package Summary

Released Continuous Integration Documented

This contains CvBridge, which converts between ROS Image messages and OpenCV images.

- Maintainer status: maintained
- Maintainer: Vincent Rabaud <vincent.rabaud AT gmail DOT com>
- Author: Patrick Mihelich, James Bowman
- License: BSD
- Bug / feature tracker: https://github.com/ros-perception/vision_opencv/issues
- Source: git https://github.com/ros-perception/vision_opencv.git (branch: kinetic)

Package Links

[Code API](#)

[Tutorials](#)

[FAQ](#)

[Changelog](#)

[Change List](#)

[Reviews](#)

Dependencies (4)

[Used by \(128\)](#)

[Jenkins jobs \(10\)](#)

Wiki

[Distributions](#)

[ROS/Installation](#)

[ROS/Tutorials](#)

[RecentChanges](#)

[cv_bridge](#)

Pagina

[Pagina non alterabile](#)

[Informazioni](#)

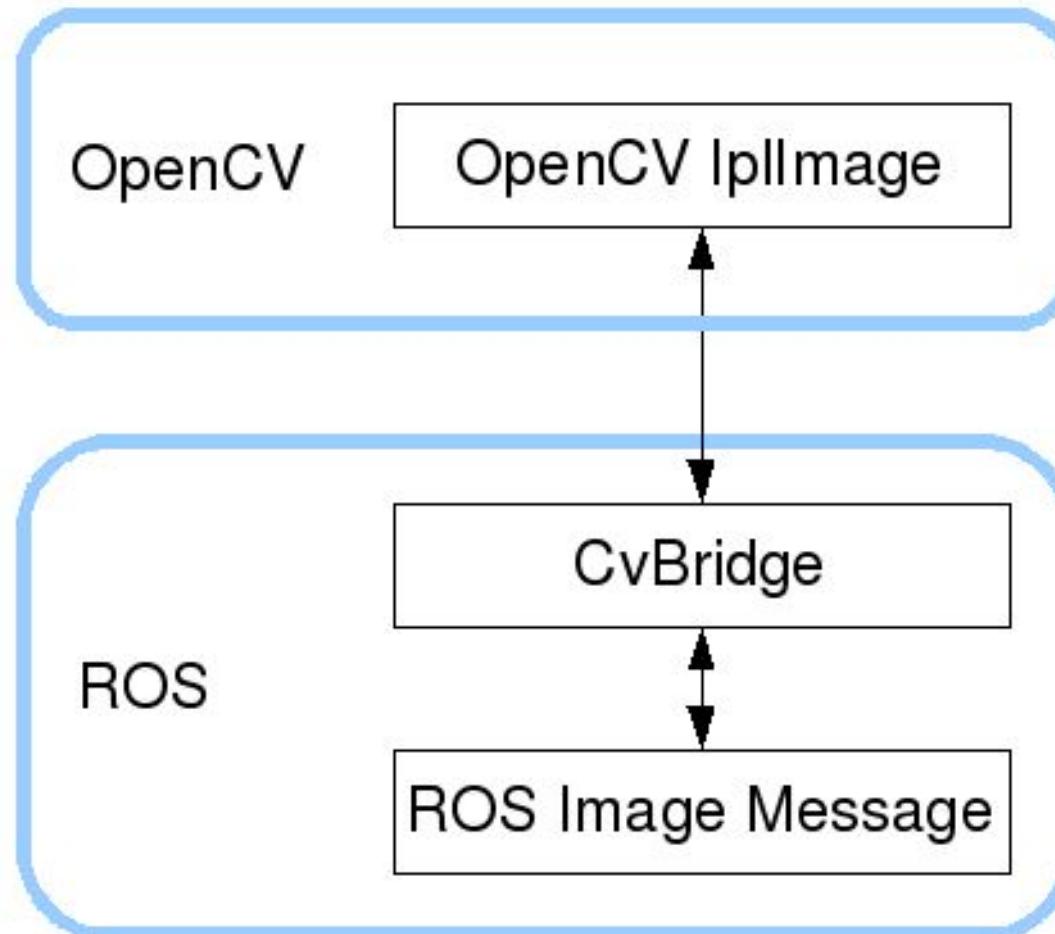
[Allegati](#)

Altre azioni:

Utente

[Accedi](#)

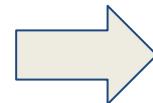
cv_bridge



Esempio: unibas_viewer

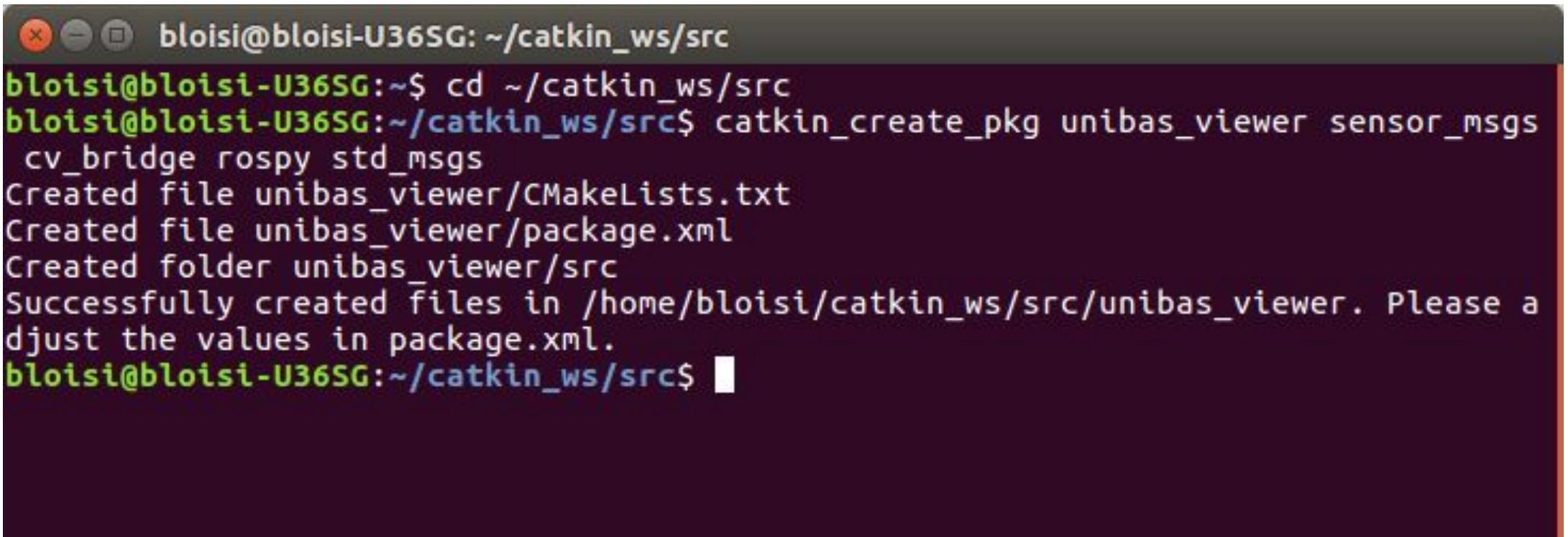


rosbag acquisita con
un sensore RGBD



visualizzazione
immagine OpenCV

creazione del nodo unibas_viewer



```
bloisi@bloisi-U36SG: ~/catkin_ws/src
bloisi@bloisi-U36SG:~$ cd ~/catkin_ws/src
bloisi@bloisi-U36SG:~/catkin_ws/src$ catkin_create_pkg unibas_viewer sensor_msgs
cv_bridge rospy std_msgs
Created file unibas_viewer/CMakeLists.txt
Created file unibas_viewer/package.xml
Created folder unibas_viewer/src
Successfully created files in /home/bloisi/catkin_ws/src/unibas_viewer. Please adjust the values in package.xml.
bloisi@bloisi-U36SG:~/catkin_ws/src$
```

cartella unibas_viewer



catkin_make

```
x - ⊞ bloisi@bloisi-U36SG: ~/catkin_ws
bloisi@bloisi-U36SG:~$ cd ~/catkin_ws/src
bloisi@bloisi-U36SG:~/catkin_ws/src$ catkin_create_pkg unibas_viewer sensor_msgs
cv_bridge rospy std_msgs
Created file unibas_viewer/CMakeLists.txt
Created file unibas_viewer/package.xml
Created folder unibas_viewer/src
Successfully created files in /home/bloisi/catkin_ws/src/unibas_viewer. Please adjust the values in package.xml.
bloisi@bloisi-U36SG:~/catkin_ws/src$ cd ..
bloisi@bloisi-U36SG:~/catkin_ws$ catkin_make
```

settiamo l'ambiente ROS

```
bloisi@bloisi-U36SG: ~/catkin_ws
[ 52%] Built target hw1_generate_messages_py
[ 53%] Built target hw1_generate_messages_lisp
. ~/catkin_ws/devel/setup.bash
[ 61%] Built target turtlebot3_applications_msgs_generate_messages_py
[ 63%] Built target turtlebot3_applications_msgs_generate_messages_cpp
[ 65%] Built target turtlebot3_applications_msgs_generate_messages_lisp
[ 70%] Built target turtlebot3_example_generate_messages_py
[ 75%] Built target turtlebot3_example_generate_messages_nodejs
[ 79%] Built target turtlebot3_example_generate_messages_cpp
[ 81%] Built target turtlebot3_applications_msgs_generate_messages_nodejs
[ 87%] Built target turtlebot3_example_generate_messages_eus
[ 89%] Built target turtlebot3_diagnostics
[ 94%] Built target turtlebot3_example_generate_messages_lisp
[ 94%] Built target turtlebot3_msgs_generate_messages
[ 96%] Built target turtlebot3_fake_node
[ 97%] Built target homework_1_generate_messages
[ 97%] Built target turtlebot3_drive
[100%] Built target turtlebot3_panorama
[100%] Built target hw1_generate_messages
[100%] Built target turtlebot3_example_generate_messages
[100%] Built target turtlebot3_applications_msgs_generate_messages
bloisi@bloisi-U36SG:~/catkin_ws$ . ~/catkin_ws/devel/setup.bash
bloisi@bloisi-U36SG:~/catkin_ws$
```

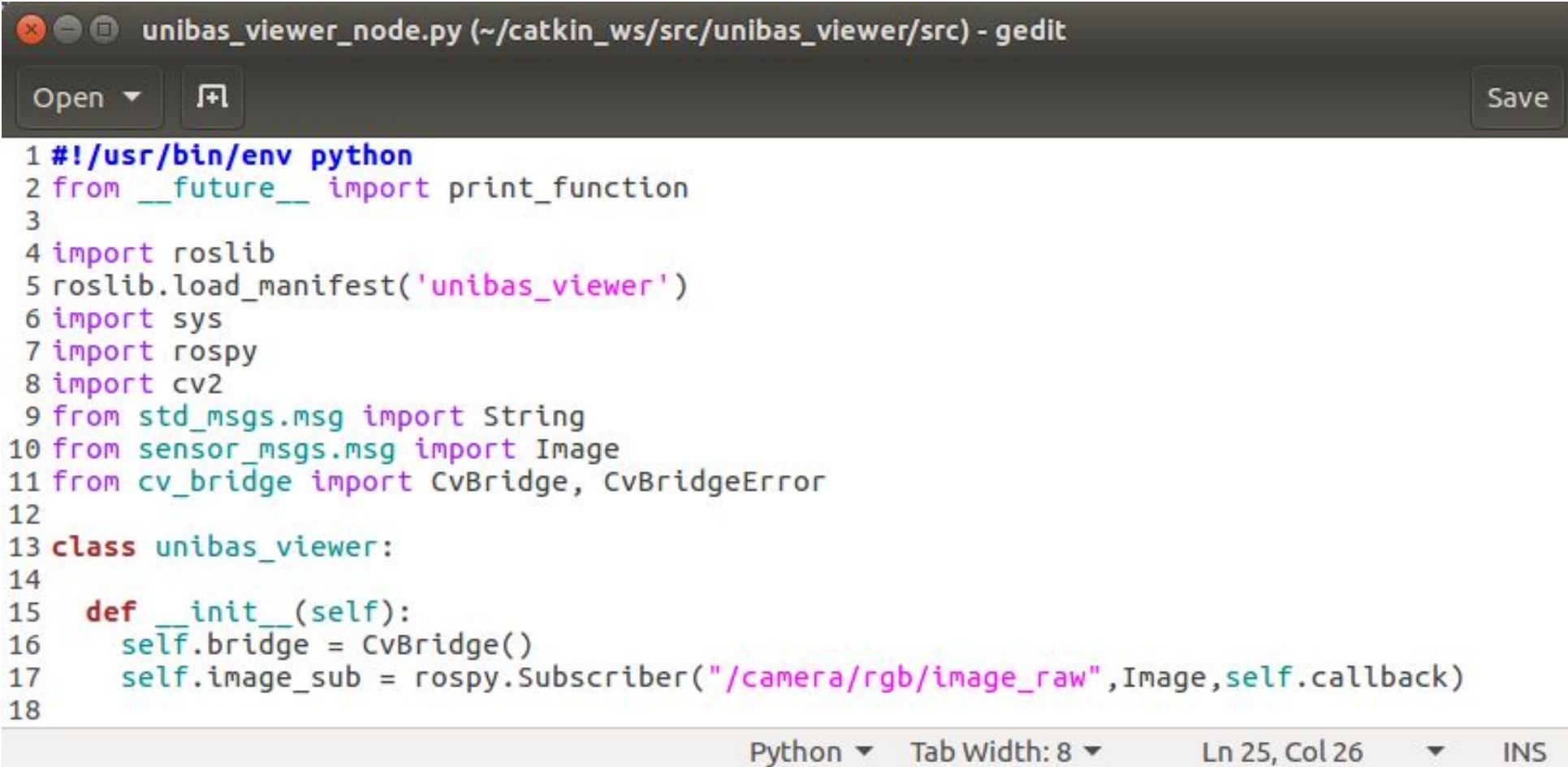
rospack find

```
bloisi@bloisi-U36SG: ~/catkin_ws
[ 58%] Built target turtlebot3_applications_msgs_generate_messages_eus
[ 61%] Built target turtlebot3_applications_msgs_generate_messages_py
[ 63%] Built target turtlebot3_applications_msgs_generate_messages_cpp
[ 65%] Built target turtlebot3_applications_msgs_generate_messages_lisp
[ 70%] Built target turtlebot3_example_generate_messages_py
[ 75%] Built target turtlebot3_example_generate_messages_nodejs
[ 79%] Built target turtlebot3_example_generate_messages_cpp
[ 81%] Built target turtlebot3_applications_msgs_generate_messages_nodejs
[ 87%] Built target turtlebot3_example_generate_messages_eus
rospack find unibas_viewer
[ 94%] Built target turtlebot3_example_generate_messages_lisp
[ 94%] Built target turtlebot3_msgs_generate_messages
[ 96%] Built target turtlebot3_fake_node
[ 97%] Built target homework_1_generate_messages
[ 97%] Built target turtlebot3_drive
[100%] Built target turtlebot3_panorama
[100%] Built target hw1_generate_messages
[100%] Built target turtlebot3_example_generate_messages
[100%] Built target turtlebot3_applications_msgs_generate_messages
bloisi@bloisi-U36SG:~/catkin_ws$ . ~/catkin_ws/devel/setup.bash
bloisi@bloisi-U36SG:~/catkin_ws$ rospack find unibas_viewer
/home/bloisi/catkin_ws/src/unibas_viewer
```

creiamo unibas_viewer_node.py



creiamo unibas_viewer_node.py

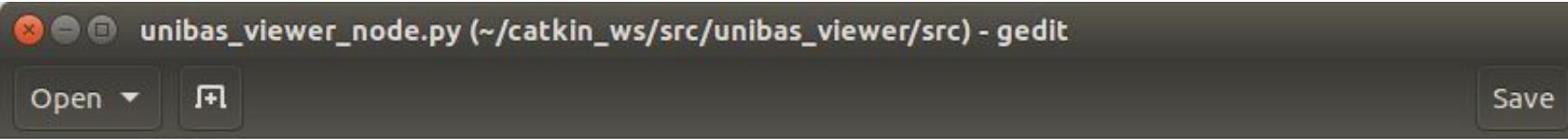


The screenshot shows a Gedit text editor window with the title "unibas_viewer_node.py (~/catkin_ws/src/unibas_viewer/src) - gedit". The code is a Python script for a ROS node named "unibas_viewer". It imports necessary modules like roslib, sys, rospy, cv2, std_msgs.msg, sensor_msgs.msg, and cv_bridge. It defines a class "unibas_viewer" with an __init__ method that initializes a CvBridge object and subscribes to the "/camera/rgb/image_raw" topic to process images.

```
1 #!/usr/bin/env python
2 from __future__ import print_function
3
4 import roslib
5 roslib.load_manifest('unibas_viewer')
6 import sys
7 import rospy
8 import cv2
9 from std_msgs.msg import String
10 from sensor_msgs.msg import Image
11 from cv_bridge import CvBridge, CvBridgeError
12
13 class unibas_viewer:
14
15     def __init__(self):
16         self.bridge = CvBridge()
17         self.image_sub = rospy.Subscriber("/camera/rgb/image_raw",Image,self.callback)
18
```

Python ▾ Tab Width: 8 ▾ Ln 25, Col 26 ▾ INS

creiamo unibas_viewer_node.py



```
unibas_viewer_node.py (~/catkin_ws/src/unibas_viewer/src) - gedit
Open ▾ Save
18
19 def callback(self,data):
20     try:
21         cv_image = self.bridge.imgmsg_to_cv2(data, "bgr8")
22     except CvBridgeError as e:
23         print(e)
24
25     cv2.imshow("image_row", cv_image)
26     cv2.waitKey(30)
27
28
29 def main(args):
30     uv = unibas_viewer()
31     rospy.init_node('unibas_viewer', anonymous=True)
32     try:
33         rospy.spin()
34     except KeyboardInterrupt:
35         print("Shutting down")
36     cv2.destroyAllWindows()
37
38 if __name__ == '__main__':
39     main(sys.argv)
40
```

Python ▾ Tab Width: 8 ▾ Ln 17, Col 75 ▾ INS

permessi per unibas_viewer_node.py

```
bloisi@bloisi-U36SG: ~/catkin_ws/src/unibas_viewer/src
bloisi@bloisi-U36SG:~/catkin_ws/
bloisi@bloisi-U36SG:~/catkin_ws$ cd src/
bloisi@bloisi-U36SG:~/catkin_ws/src$ cd unibas_viewer/
bloisi@bloisi-U36SG:~/catkin_ws/src/unibas_viewer$ cd src/
bloisi@bloisi-U36SG:~/catkin_ws/src/unibas_viewer/src$ ls
bloisi@bloisi-U36SG:~/catkin_ws/src/unibas_viewer/src$ chmod +x unibas_viewer_node.py
bloisi@bloisi-U36SG:~/catkin_ws/src/unibas_viewer/src$ █
```

roscore

```
roscore http://localhost:11311/
bloisi@bloisi-U36SG:~$ roscore
... logging to /home/bloisi/.ros/log/78cf387c-7bbf-11e9-b0ad-50465dde6884/roslau
nch-bloisi-U36SG-8561.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://localhost:35105/
ros_comm version 1.12.14

SUMMARY
=====

PARAMETERS
* /rosdistro: kinetic
* /rosversion: 1.12.14

NODES

auto-starting new master
process[master]: started with pid [8584]
ROS_MASTER_URI=http://localhost:11311/

setting /run_id to 78cf387c-7bbf-11e9-b0ad-50465dde6884
process[rosout-1]: started with pid [8733]
started core service [/rosout]
```

rosrun

```
bloisi@bloisi-U36SG:~$ rosrun unibas_viewer unibas_viewer_node.py
```

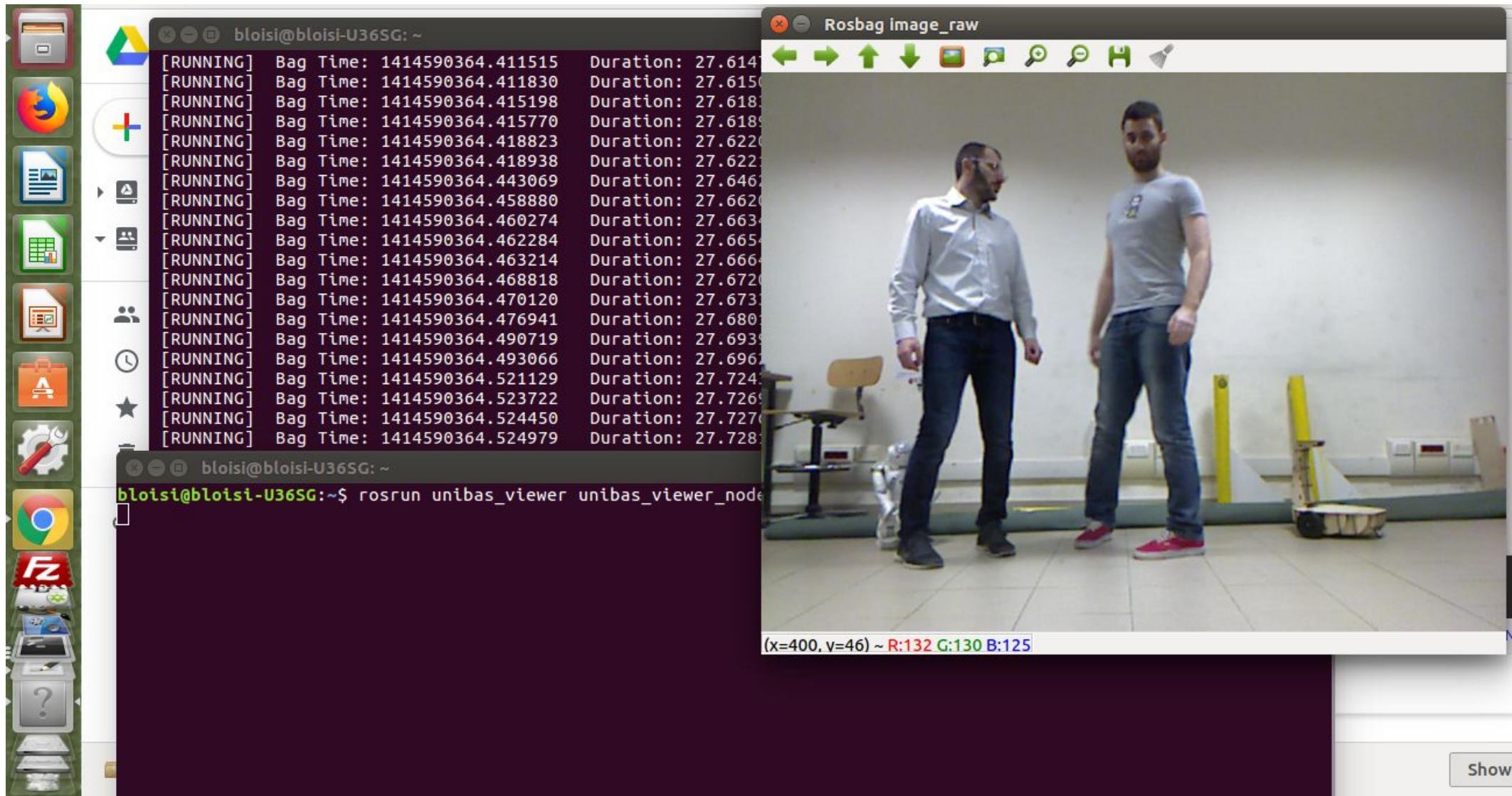
rosplay



A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains the command "rosbag play ~/Desktop/people.bag" in green text, indicating it has been typed but not yet executed. The background of the terminal is dark.

```
bloisi@bloisi-U36SG:~$ rosbag play ~/Desktop/people.bag
```

visualizzazione



rostopic list

```
bloisi@bloisi-U36SG:~$ rostopic list
/camera/debayer/parameter_descriptions
/camera/debayer/parameter_updates
/camera/depth/image/compressed/parameter_descriptions
/camera/depth/image/compressed/parameter_updates
/camera/depth/image/compressedDepth/parameter_descriptions
/camera/depth/image/compressedDepth/parameter_updates
/camera/depth/image/theora/parameter_descriptions
/camera/depth/image/theora/parameter_updates
/camera/depth/image_raw/compressed/parameter_descriptions
/camera/depth/image_raw/compressed/parameter_updates
/camera/depth/image_raw/compressedDepth/parameter_descriptions
/camera/depth/image_raw/compressedDepth/parameter_updates
/camera/depth/image_raw/theora/parameter_descriptions
/camera/depth/image_raw/theora/parameter_updates
/camera/depth/image_rect/compressed/parameter_descriptions
/camera/depth/image_rect/compressed/parameter_updates
/camera/depth/image_rect/compressedDepth/parameter_descriptions
/camera/depth/image_rect/compressedDepth/parameter_updates
/camera/depth/image_rect/theora/parameter_descriptions
/camera/depth/image_rect/theora/parameter_updates
/camera/depth/image_rect_raw/compressed/parameter_descriptions
/camera/depth/image_rect_raw/compressed/parameter_updates
/camera/depth/image_rect_raw/compressedDepth/parameter_descriptions
/camera/depth/image_rect_raw/compressedDepth/parameter_updates
/camera/depth/image_rect_raw/theora/parameter_descriptions
/camera/depth/image_rect_raw/theora/parameter_updates
/camera/depth_rectify_depth/parameter_descriptions
/camera/depth_rectify_depth/parameter_updates
/camera/depth_registered/camera_info
/camera/depth_registered/disparity
/camera/depth_registered/hw_registered/image_rect_raw
/camera/depth_registered/hw_registered/image_rect_raw/compressed
/camera/depth_registered/hw_registered/image_rect_raw/compressed/parameter_descriptions
/camera/depth_registered/hw_registered/image_rect_raw/compressed/parameter_updates
/camera/depth_registered/hw_registered/image_rect_raw/compressedDepth
/camera/depth_registered/hw_registered/image_rect_raw/compressedDepth/parameter_descriptions
/camera/depth_registered/hw_registered/image_rect_raw/compressedDepth/parameter_updates
/camera/depth_registered/hw_registered/image_rect_raw/theora
```

rostopic list

```
bloisi@bloisi-U36SG: ~
/camera/depth_registered/hw_registered/image_rect_raw/theora/parameter_descriptions
/camera/depth_registered/hw_registered/image_rect_raw/theora/parameter_updates
/camera/depth_registered/image_raw
/camera/depth_registered/image_raw/compressed
/camera/depth_registered/image_raw/compressed/parameter_descriptions
/camera/depth_registered/image_raw/compressed/parameter_updates
/camera/depth_registered/image_raw/compressedDepth
/camera/depth_registered/image_raw/compressedDepth/parameter_descriptions
/camera/depth_registered/image_raw/compressedDepth/parameter_updates
/camera/depth_registered/image_raw/theora
/camera/depth_registered/image_raw/theora/parameter_descriptions
/camera/depth_registered/image_raw/theora/parameter_updates
/camera/depth_registered/points
/camera/depth_registered/sw_registered/image_rect_raw/compressed/parameter_descriptions
/camera/depth_registered/sw_registered/image_rect_raw/compressed/parameter_updates
/camera/depth_registered/sw_registered/image_rect_raw/compressedDepth/parameter_descriptions
/camera/depth_registered/sw_registered/image_rect_raw/compressedDepth/parameter_updates
/camera/depth_registered/sw_registered/image_rect_raw/theora/parameter_descriptions
/camera/depth_registered/sw_registered/image_rect_raw/theora/parameter_updates
/camera/depth_registered_rectify_depth/parameter_descriptions
/camera/depth_registered_rectify_depth/parameter_updates
/camera/driver/parameter_descriptions
/camera/driver/parameter_updates
/camera/ir/image_raw/compressed/parameter_descriptions
/camera/ir/image_raw/compressed/parameter_updates
/camera/ir/image_raw/compressedDepth/parameter_descriptions
/camera/ir/image_raw/compressedDepth/parameter_updates
/camera/ir/image_raw/theora/parameter_descriptions
/camera/ir/image_raw/theora/parameter_updates
/camera/ir/image_rect_ir/compressed/parameter_descriptions
/camera/ir/image_rect_ir/compressed/parameter_updates
/camera/ir/image_rect_ir/compressedDepth/parameter_descriptions
/camera/ir/image_rect_ir/compressedDepth/parameter_updates
/camera/ir/image_rect_ir/theora/parameter_descriptions
/camera/ir/image_rect_ir/theora/parameter_updates
/camera/projector/camera_info
/camera/rectify_color/parameter_descriptions
/camera/rectify_color/parameter_updates
/camera/rectify_ir/parameter_descriptions
```

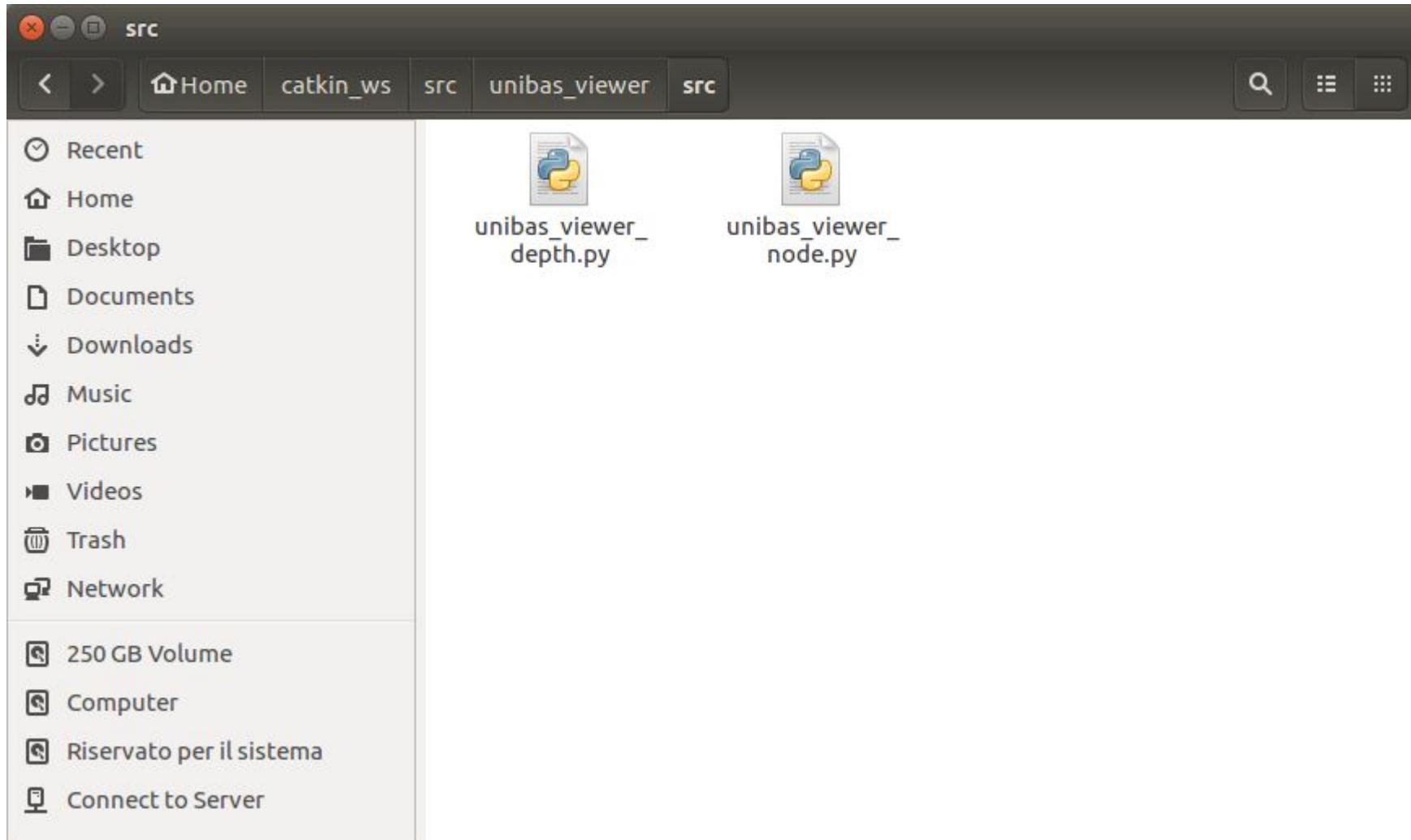
rostopic list

```
bloisi@bloisi-U36SG: ~
/camera/rgb/image_color/theora/parameter_updates
/camera/rgb/image_mono
/camera/rgb/image_mono/compressed
/camera/rgb/image_mono/compressed/parameter_descriptions
/camera/rgb/image_mono/compressed/parameter_updates
/camera/rgb/image_mono/compressedDepth/parameter_descriptions
/camera/rgb/image_mono/compressedDepth/parameter_updates
/camera/rgb/image_mono/theora
/camera/rgb/image_mono/theora/parameter_descriptions
/camera/rgb/image_mono/theora/parameter_updates
/camera/rgb/image_raw
/camera/rgb/image_raw/compressed
/camera/rgb/image_raw/compressed/parameter_descriptions
/camera/rgb/image_raw/compressed/parameter_updates
/camera/rgb/image_raw/compressedDepth/parameter_descriptions
/camera/rgb/image_raw/compressedDepth/parameter_updates
/camera/rgb/image_raw/theora
/camera/rgb/image_raw/theora/parameter_descriptions
/camera/rgb/image_raw/theora/parameter_updates
/camera/rgb/image_rect_color
/camera/rgb/image_rect_color/compressed
/camera/rgb/image_rect_color/compressed/parameter_descriptions
/camera/rgb/image_rect_color/compressed/parameter_updates
/camera/rgb/image_rect_color/compressedDepth/parameter_descriptions
/camera/rgb/image_rect_color/compressedDepth/parameter_updates
/camera/rgb/image_rect_color/theora
/camera/rgb/image_rect_color/theora/parameter_descriptions
/camera/rgb/image_rect_color/theora/parameter_updates
/camera/rgb/image_rect_mono
/camera/rgb/image_rect_mono/compressed
/camera/rgb/image_rect_mono/compressed/parameter_descriptions
/camera/rgb/image_rect_mono/compressed/parameter_updates
/camera/rgb/image_rect_mono/compressedDepth/parameter_descriptions
/camera/rgb/image_rect_mono/compressedDepth/parameter_updates
/camera/rgb/image_rect_mono/theora
/camera/rgb/image_rect_mono/theora/parameter_descriptions
/camera/rgb/image_rect_mono/theora/parameter_updates
/clock
```

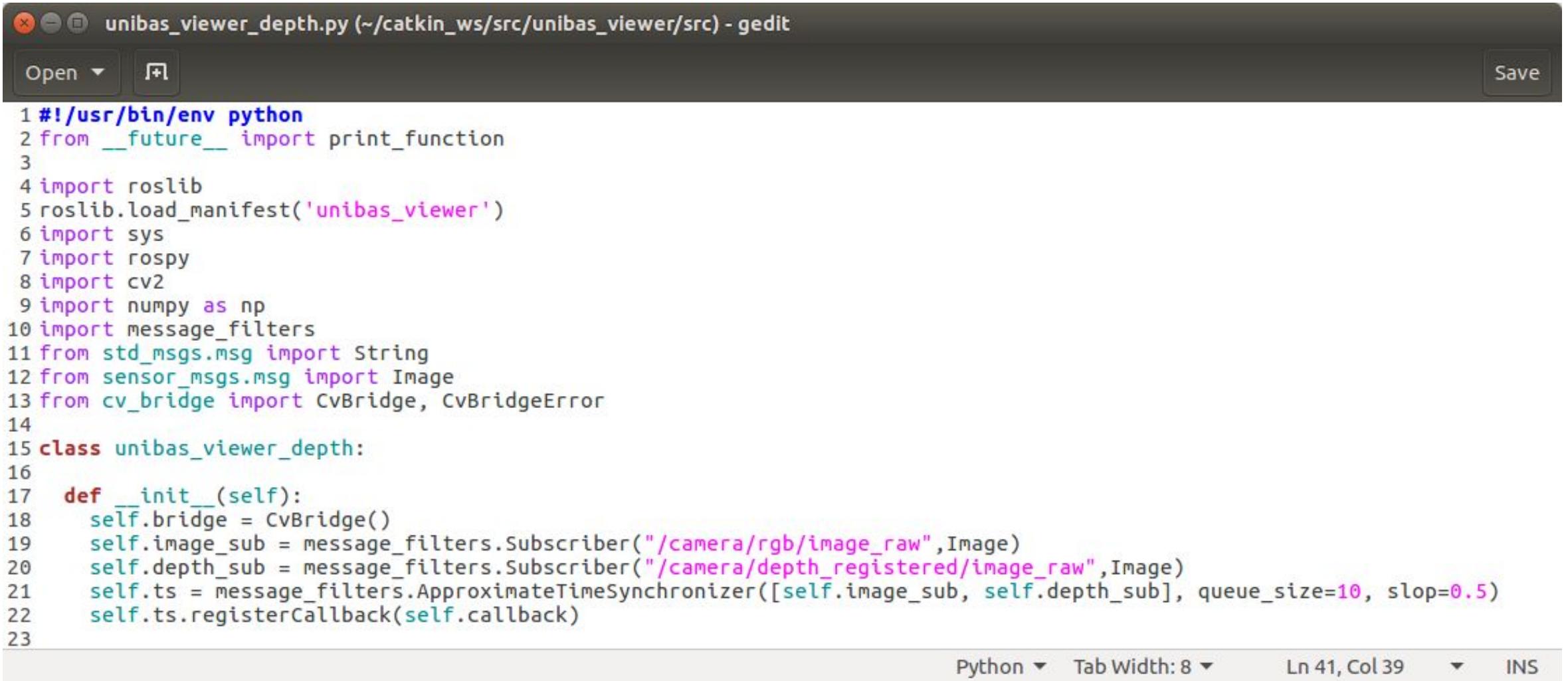
rostopic list

```
bloisi@bloisi-U36SG: ~
/camera/rgb/image_color/theora/parameter_updates
/camera/rgb/image_mono
/camera/rgb/image_mono/compressed
/camera/rgb/image_mono/compressed/parameter_descriptions
/camera/rgb/image_mono/compressed/parameter_updates
/camera/rgb/image_mono/compressedDepth/parameter_descriptions
/camera/rgb/image_mono/compressedDepth/parameter_updates
/camera/rgb/image_mono/theora
/camera/rgb/image_mono/theora/parameter_descriptions
/camera/rgb/image_mono/theora/parameter_updates
/camera/rgb/image_raw
/camera/rgb/image_raw/compressed
/camera/rgb/image_raw/compressed/parameter_descriptions
/camera/rgb/image_raw/compressed/parameter_updates
/camera/rgb/image_raw/compressedDepth/parameter_descriptions
/camera/rgb/image_raw/compressedDepth/parameter_updates
/camera/rgb/image_raw/theora
/camera/rgb/image_raw/theora/parameter_descriptions
/camera/rgb/image_raw/theora/parameter_updates
/camera/rgb/image_rect_color
/camera/rgb/image_rect_color/compressed
/camera/rgb/image_rect_color/compressed/parameter_descriptions
/camera/rgb/image_rect_color/compressed/parameter_updates
/camera/rgb/image_rect_color/compressedDepth/parameter_descriptions
/camera/rgb/image_rect_color/compressedDepth/parameter_updates
/camera/rgb/image_rect_color/theora
/camera/rgb/image_rect_color/theora/parameter_descriptions
/camera/rgb/image_rect_color/theora/parameter_updates
/camera/rgb/image_rect_mono
/camera/rgb/image_rect_mono/compressed
/camera/rgb/image_rect_mono/compressed/parameter_descriptions
/camera/rgb/image_rect_mono/compressed/parameter_updates
/camera/rgb/image_rect_mono/compressedDepth/parameter_descriptions
/camera/rgb/image_rect_mono/compressedDepth/parameter_updates
/camera/rgb/image_rect_mono/theora
/camera/rgb/image_rect_mono/theora/parameter_descriptions
/camera/rgb/image_rect_mono/theora/parameter_updates
/clock
```

visualizziamo la depth



unibas_viewer_node.py



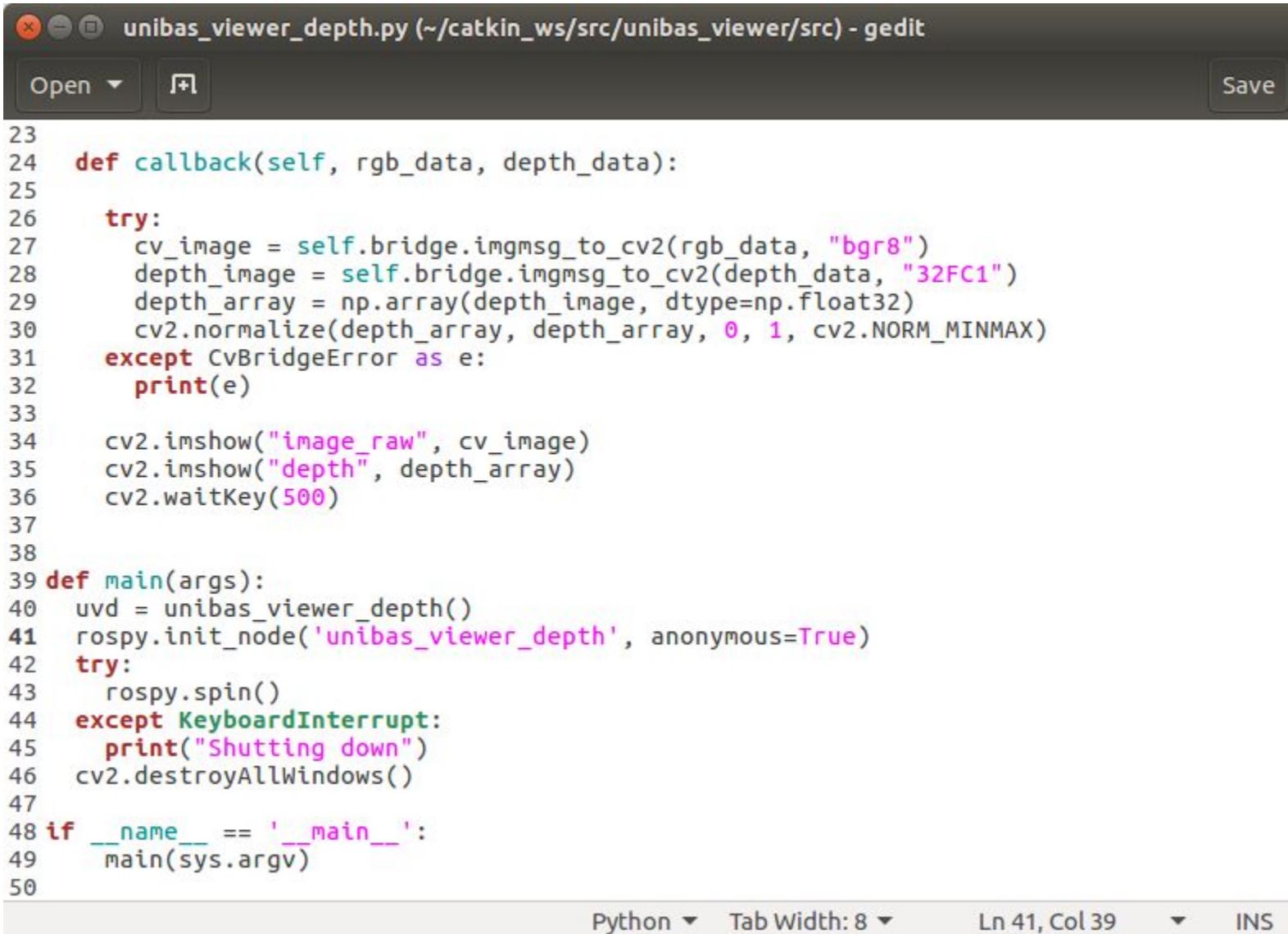
The screenshot shows a Gedit text editor window with the title "unibas_viewer_depth.py (~/catkin_ws/src/unibas_viewer/src) - gedit". The code editor displays a Python script for a ROS node. The script imports various ROS and cv_bridge modules and defines a class "unibas_viewer_depth" with an __init__ method that sets up subscribers for camera image and depth registered image topics, and an ApproximateTimeSynchronizer to synchronize them.

```
1 #!/usr/bin/env python
2 from __future__ import print_function
3
4 import roslib
5 roslib.load_manifest('unibas_viewer')
6 import sys
7 import rospy
8 import cv2
9 import numpy as np
10 import message_filters
11 from std_msgs.msg import String
12 from sensor_msgs.msg import Image
13 from cv_bridge import CvBridge, CvBridgeError
14
15 class unibas_viewer_depth:
16
17     def __init__(self):
18         self.bridge = CvBridge()
19         self.image_sub = message_filters.Subscriber("/camera/rgb/image_raw",Image)
20         self.depth_sub = message_filters.Subscriber("/camera/depth_registered/image_raw",Image)
21         self.ts = message_filters.ApproximateTimeSynchronizer([self.image_sub, self.depth_sub], queue_size=10, slop=0.5)
22         self.ts.registerCallback(self.callback)
23
```

Python ▾ Tab Width: 8 ▾ Ln 41, Col 39 ▾ INS

the slop parameter defines the delay (in seconds) with which messages can be synchronized.

unibas_viewer_node.py

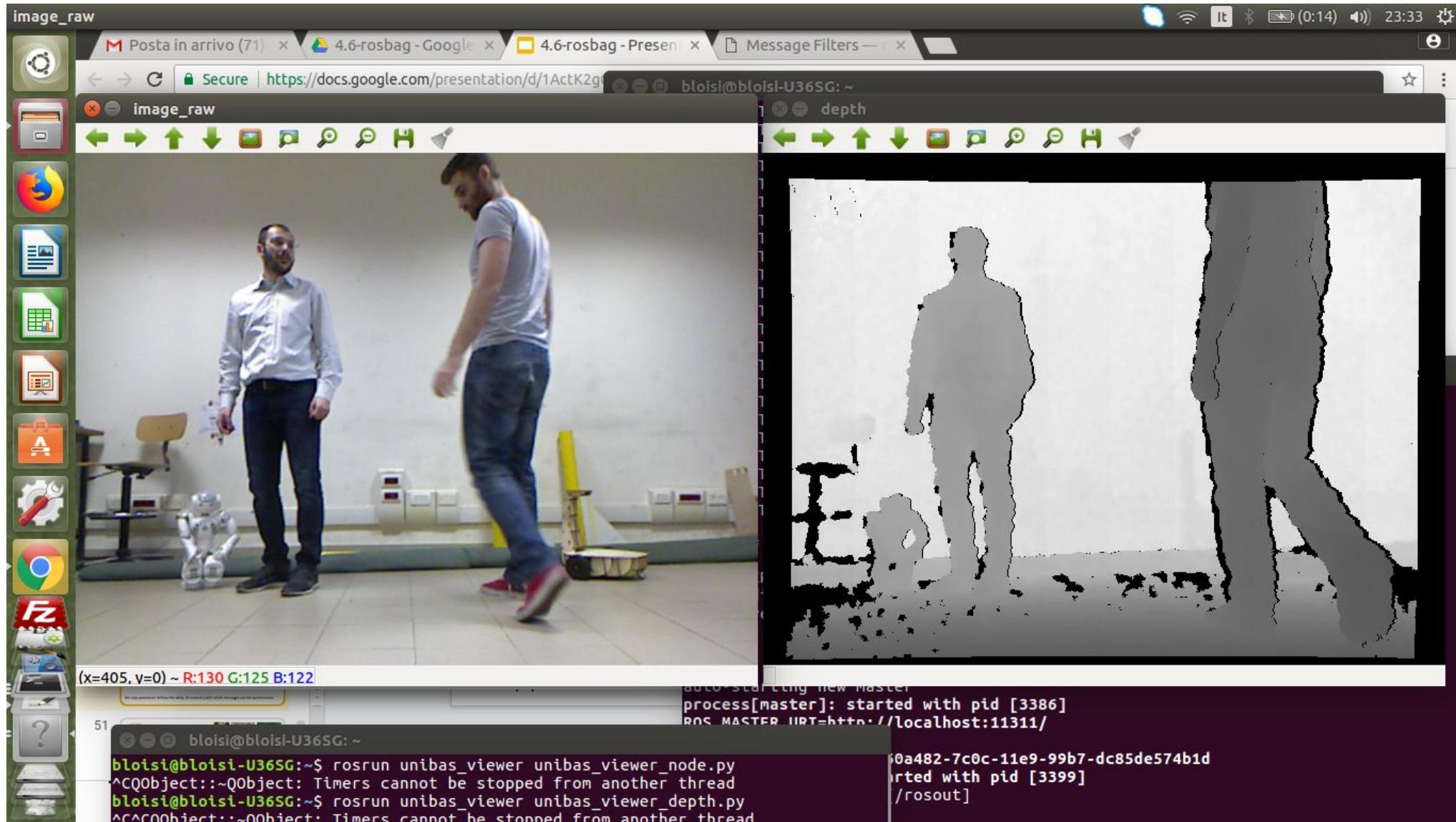


The screenshot shows a Gedit text editor window with the title "unibas_viewer_depth.py (~/catkin_ws/src/unibas_viewer/src) - gedit". The code is a Python script for a ROS node named 'unibas_viewer_depth'. It uses the cv2 library to process RGB and depth sensor data. The code includes a callback function for sensor data and a main function for node initialization and shutdown. The code is color-coded with syntax highlighting for keywords, comments, and variable names.

```
23
24 def callback(self, rgb_data, depth_data):
25
26     try:
27         cv_image = self.bridge.imgmsg_to_cv2(rgb_data, "bgr8")
28         depth_image = self.bridge.imgmsg_to_cv2(depth_data, "32FC1")
29         depth_array = np.array(depth_image, dtype=np.float32)
30         cv2.normalize(depth_array, depth_array, 0, 1, cv2.NORM_MINMAX)
31     except CvBridgeError as e:
32         print(e)
33
34     cv2.imshow("image_raw", cv_image)
35     cv2.imshow("depth", depth_array)
36     cv2.waitKey(500)
37
38
39 def main(args):
40     uvd = unibas_viewer_depth()
41     rospy.init_node('unibas_viewer_depth', anonymous=True)
42     try:
43         rospy.spin()
44     except KeyboardInterrupt:
45         print("Shutting down")
46     cv2.destroyAllWindows()
47
48 if __name__ == '__main__':
49     main(sys.argv)
50
```

Python Tab Width: 8 Ln 41, Col 39 INS

unibas_viewer_node.py



repository unibas_viewer

Il codice del repository unibas_viewer è disponibile al seguente link

https://github.com/dbloisi/unibas_viewer

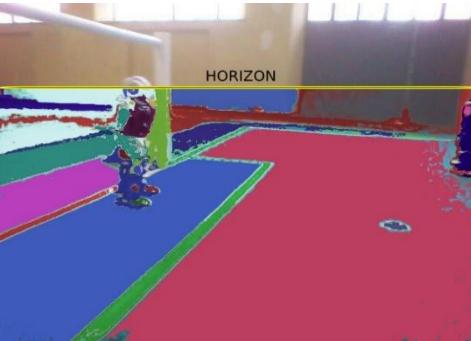
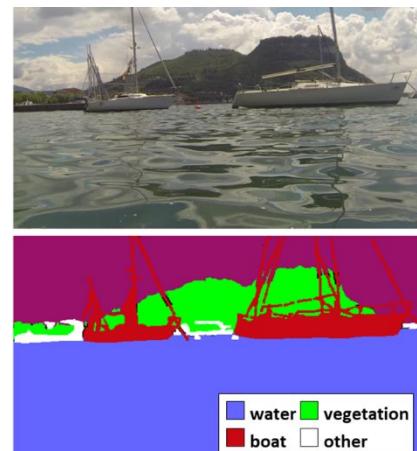
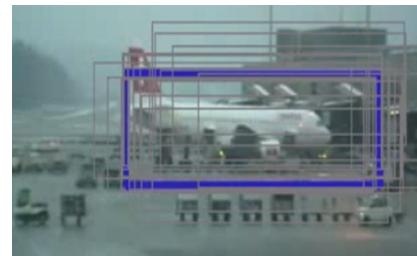
The screenshot shows a GitHub repository page for 'dbloisi/unibas_viewer'. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. The repository name 'dbloisi / unibas_viewer' is displayed above a summary bar. This bar shows 1 unwatched star, 0 forks, and 0 issues. Below the summary, there are tabs for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. A brief description of the repository is provided: 'ros node for reading a rosbag containing rgbd data'. There are 'Edit' and 'Manage topics' buttons at the bottom of this section.



UNIVERSITÀ DEGLI STUDI DELLA BASILICATA

Corso di Sistemi Informativi
A.A. 2018/19

Docente
Domenico Daniele Bloisi



Maggio 2019

water vegetation
boat other