



UNIVERSITÀ
di VERONA

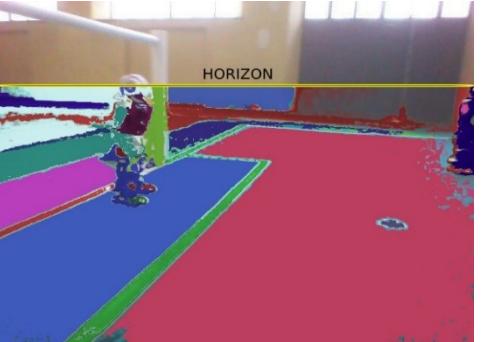
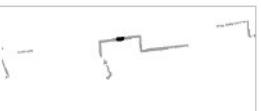
Dipartimento
di INFORMATICA

*Corso di Laboratorio Ciberfisico
Modulo di Robot Programming with ROS*

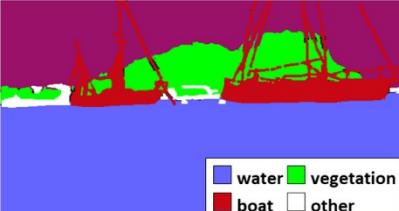
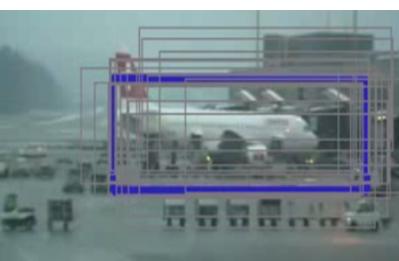
Esercitazione

Face detection

Maggio 2018

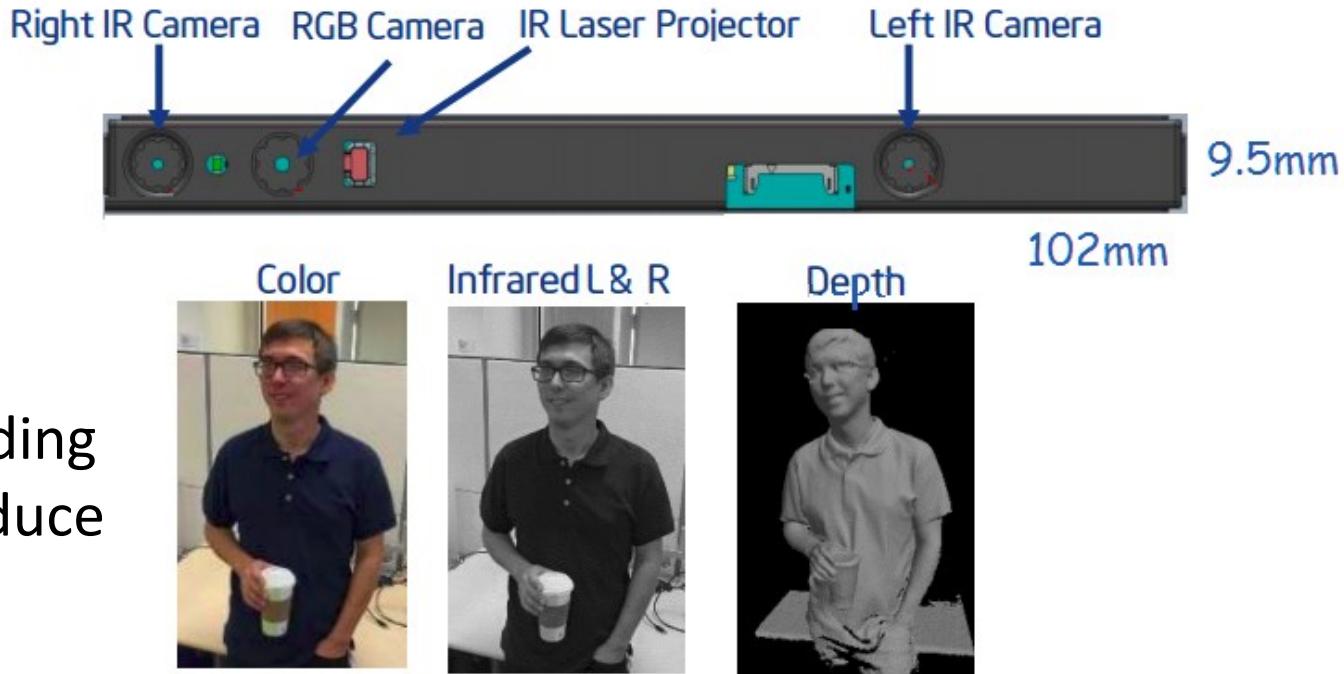
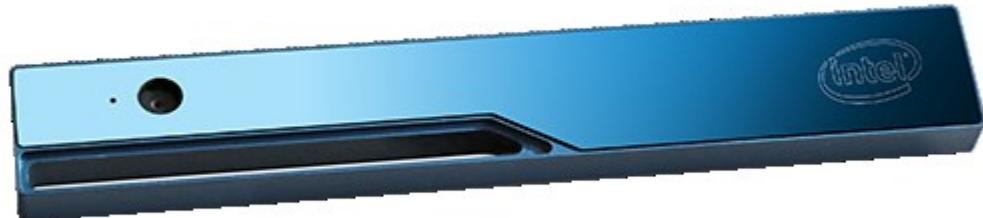


Docente:
Domenico Daniele
Bloisi



■ water ■ vegetation
■ boat ■ other

Intel RealSense Camera R200



- The R200 actually has 3 cameras providing RGB (color) and stereoscopic IR to produce depth
- The inside range is approximately 0.5-3.5 meters and an outside range up to 10 meters
- With the help of a laser projector, the camera does 3D scanning for scene perception and enhanced photography

Realsense_r200_viewer

The screenshot shows a GitHub repository page for 'realsense_r200_viewer'. The repository is owned by 'labrobotica-bloisi'. A red oval highlights the repository name 'realsense_r200_viewer' in the header. Another red oval highlights the 'README.md' file entry in the file list.

Repository Information:

- Owner: labrobotica-bloisi
- Name: realsense_r200_viewer
- Code: 2 commits
- Issues: 0
- Pull requests: 0
- Projects: 0
- Wiki: 0
- Insights: 0
- Settings: 0
- Watch: 0
- Star: 0
- Fork: 0

File List:

- Branch: master
- New pull request
- Create new file
- Upload files
- Find file
- Clone or download

File	Description	Last Commit
labrobotica-bloisi Initial commit		Latest commit c3aab1e an hour ago
README.md	Initial commit	an hour ago
README.md		

README.md Content:

```
realsense_r200_viewer

ROS node for visualizing data coming from an Intel RealSense R200 device
```

clone

The screenshot shows a GitHub repository page for `labrobotica-bloisi/realsense_r200_viewer`. The page includes a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the navigation bar, there are buttons for Watch (0), Star (0), and Fork (0). The main content area displays the repository's description: "ROS node for visualizing data coming from an Intel RealSense R200 device". There are sections for 2 commits, 1 branch, 0 releases, and 1 contributor. A dropdown menu for cloning the repository is open, showing options for "Clone with HTTPS" and "Use SSH", along with the URL `https://github.com/labrobotica-bloisi/realsense_r200_viewer`. A red circle highlights the "Clone or download" button in the dropdown menu.

ROS node for visualizing data coming from an Intel RealSense R200 device

Add topics

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request

labrobotica-bloisi Initial commit

README.md Initial commit

README.md

Clone with HTTPS Use SSH
https://github.com/labrobotica-bloisi/realsense_r200_viewer

Download ZIP

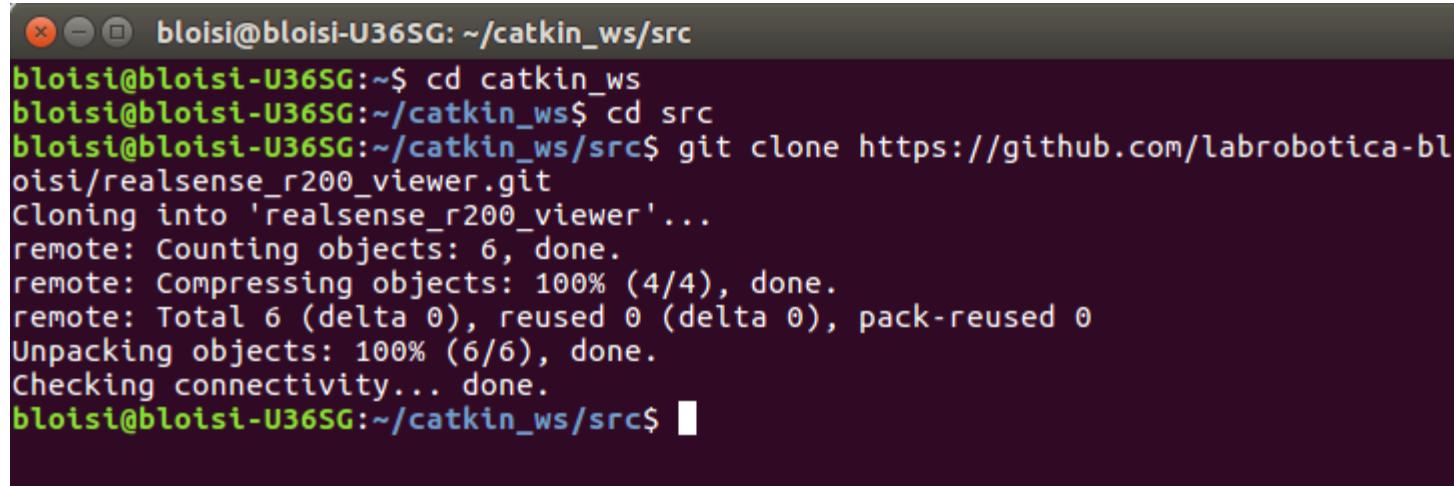
realsense_r200_viewer

ROS node for visualizing data coming from an Intel RealSense R200 device

Creazione del repository locale

Repository su GitHub

https://github.com/labrobotica-bloisi/realsense_r200_viewer



A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~/catkin_ws/src". The terminal shows the command "git clone https://github.com/labrobotica-bloisi/realsense_r200_viewer.git" being run. The output of the command is displayed, showing the progress of cloning the repository, including object counting, compression, and unpacking. The terminal window has a dark background with light-colored text.

```
bloisi@bloisi-U36SG:~/catkin_ws/src
bloisi@bloisi-U36SG:~$ cd catkin_ws
bloisi@bloisi-U36SG:~/catkin_ws$ cd src
bloisi@bloisi-U36SG:~/catkin_ws/src$ git clone https://github.com/labrobotica-bloisi/realsense_r200_viewer.git
Cloning into 'realsense_r200_viewer'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
Checking connectivity... done.
bloisi@bloisi-U36SG:~/catkin_ws/src$
```

Repository locale

creato in

~/catkin_ws/src

Package.xml

```
<?xml version="1.0"?>
<package>
  <name>realsense_r200_viewer</name>
  <version>0.0.1</version>
  <description>realsense_r200_viewer package</description>
  <maintainer email="domenico.bloisi@gmail.com">Domenico Bloisi</maintainer>
  <license>LGPLv3</license>

  <buildtool_depend>catkin</buildtool_depend>

  <build_depend>cv_bridge</build_depend>
  <build_depend>image_transport</build_depend>
  <build_depend>roscpp</build_depend>
  <build_depend>rospy</build_depend>
  <build_depend>sensor_msgs</build_depend>
  <build_depend>std_msgs</build_depend>

  <run_depend>cv_bridge</run_depend>
  <run_depend>image_transport</run_depend>
  <run_depend>roscpp</run_depend>
  <run_depend>rospy</run_depend>
  <run_depend>sensor_msgs</run_depend>
  <run_depend>std_msgs</run_depend>
</package>
```

r200_viewer.cpp

```
#include <ros/ros.h>
#include <sensor_msgs/Image.h>
#include <cv_bridge/cv_bridge.h>

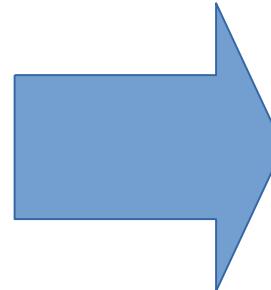
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/imgcodecs.hpp>

using namespace std;

void rgb_callback(const sensor_msgs::ImageConstPtr& msg){
    cv_bridge::CvImagePtr cv_ptr;

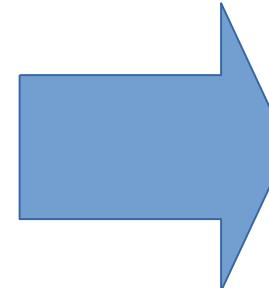
    try{
        cv_ptr = cv_bridge::toCvCopy(msg, sensor_msgs::image_encodings::BGR8);
    } catch (cv_bridge::Exception& e){
        ROS_ERROR("cv_bridge exception: %s", e.what());
        return;
    }

    cv::imshow("RGB", cv_ptr->image);
    cv::waitKey(30);
}
```



r200_viewer.cpp

```
void depth_callback(const sensor_msgs::ImageConstPtr& msg){  
    cv_bridge::CvImagePtr cv_ptr;  
  
    try {  
        cv_ptr = cv_bridge::toCvCopy(msg, msg->encoding);  
    } catch (cv_bridge::Exception& e) {  
        ROS_ERROR("cv_bridge exception: %s", e.what());  
        return;  
    }  
  
    cv::Mat depthMat = cv_ptr->image.clone();  
  
    double dmin, dmax;  
    cv::minMaxIdx(depthMat, &dmin, &dmax);  
    cv::Mat adjMat;  
    cv::convertScaleAbs(depthMat, adjMat, 255 / dmax);  
  
    cv::Mat colorMat;  
    cv::applyColorMap(adjMat, colorMat, cv::COLORMAP_HOT);  
    cv::imshow("Depth", colorMat);  
  
    cv::waitKey(30);  
}
```



r200_viewer.cpp

```
int main(int argc, char **argv)
{
    ros::init(argc, argv, "realsense_r200_viewer");

    ros::NodeHandle nh;

    string topic_rgb = "camera/rgb/image_rect_color";
    string topic_depth = "camera/depth_registered/sw_registered/image_rect_raw";

    cout << "Subscriptions:" << endl;
    cout << " - RGB topic: " << topic_rgb << endl;
    cout << " - Depth topic: " << topic_depth << endl;

    ros::Subscriber rgb_sub = nh.subscribe(topic_rgb, 1, &rgb_callback);
    ros::Subscriber depth_sub = nh.subscribe(topic_depth, 1, &depth_callback);

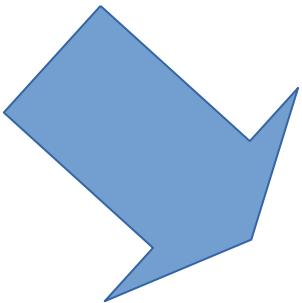
    ros::spin();

    return 0;
}
```

catkin_make

catkin_make --pkg realsense_r200_viewer

```
bloisi@bloisi-U36SG:~/catkin_ws
bloisi@bloisi-U36SG:~/catkin_ws$ catkin_make --pkg realsense_r200_viewer
```

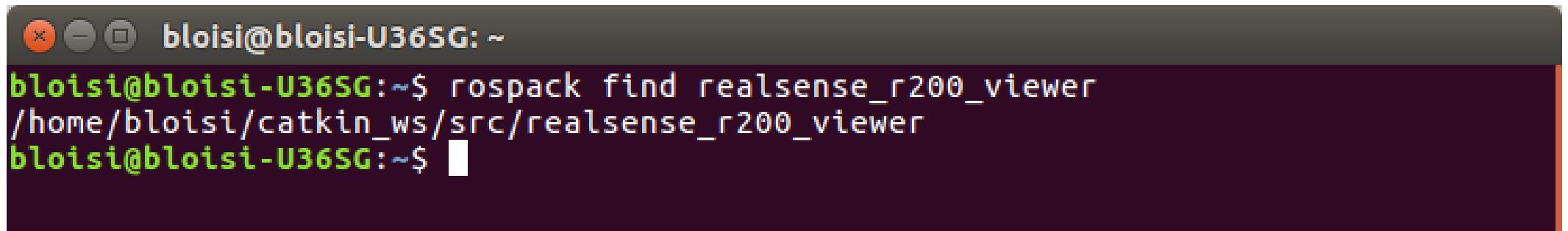


```
bloisi@bloisi-U36SG:~/catkin_ws
bloisi@bloisi-U36SG:~/catkin_ws$ catkin_make --pkg realsense_r200_viewer
Base path: /home/bloisi/catkin_ws
Source space: /home/bloisi/catkin_ws/src
Build space: /home/bloisi/catkin_ws/build
Devel space: /home/bloisi/catkin_ws/devel
Install space: /home/bloisi/catkin_ws/install
#####
##### Running command: "make cmake_check_build_system" in "/home/bloisi/catkin_ws/build"
#####
#####
##### Running command: "make -j4 -l4" in "/home/bloisi/catkin_ws/build/realsense_r200_viewer"
#####
[100%] Built target realsense_r200_viewer
bloisi@bloisi-U36SG:~/catkin_ws$
```

Finding a ROS package

Now that your package has a manifest, ROS can find it. Try executing the command:

```
rospack find realsense_r200_viewer
```

A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains the command "rospack find realsense_r200_viewer" followed by its output: "/home/bloisi/catkin_ws/src/realsense_r200_viewer".

```
bloisi@bloisi-U36SG:~$ rospack find realsense_r200_viewer
/home/bloisi/catkin_ws/src/realsense_r200_viewer
bloisi@bloisi-U36SG:~$
```

if ROS is set up correctly you should see the physical location where your package is stored

roscore + rosrun

Apriamo un terminale e lanciamo roscore

```
roscore http://localhost:11311/
bloisi@bloisi-U36SG:~$ roscore
... logging to /home/bloisi/.ros/log/ff2cc138-d525-11e7-a75c-dc85de574b1d/roslau
nch-bloisi-U36SG-21837.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:40525/
ros_comm version 1.12.7

SUMMARY
=====
PARAMETERS
 * /rostdistro: kinetic
 * /rosversion: 1.12.7

NODES
auto-starting new master
process[master]: started with pid [21848]
ROS_MASTER_URI=http://localhost:11311/
setting /run_id to ff2cc138-d525-11e7-a75c-dc85de574b1d
process[rosout-1]: started with pid [21861]
started core service [/rosout]

bloisi@bloisi-U36SG-21837:~$
```

Apriamo un secondo terminale e lanciamo

```
rosrun realsense_r200_viewer  
realsense r200 viewer
```

```
bloisi@bloisi-U36SG:~/catkin_ws$ rosrun realsense_r200_viewer realsense_r200_viewer
Subscriptions:
- RGB topic: camera/rgb/image_rect_color
- Depth topic: camera/depth_registered/sw_registered/image_rect_raw
```

Cosa accade?

Sensore reale o bag file

Abbiamo bisogno di dati in ingresso per poterli visualizzare

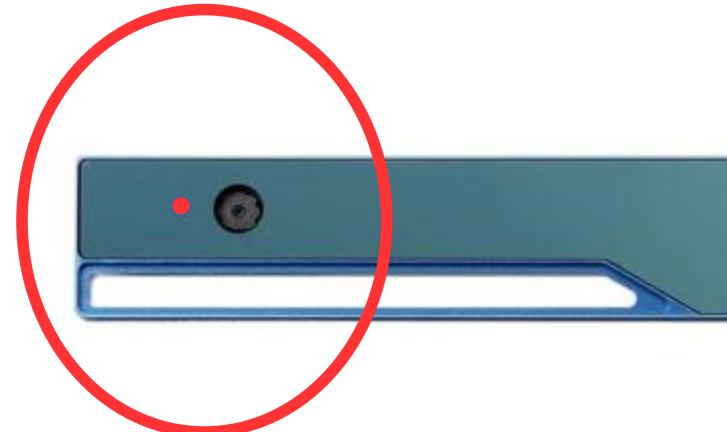
- i dati vengono acquisiti tramite il sensore reale
- utilizziamo i dati registrati contenuti in un bag file

realsense_camera node

```
roslaunch realsense_camera r200_nodelet_rgbd.launch
```

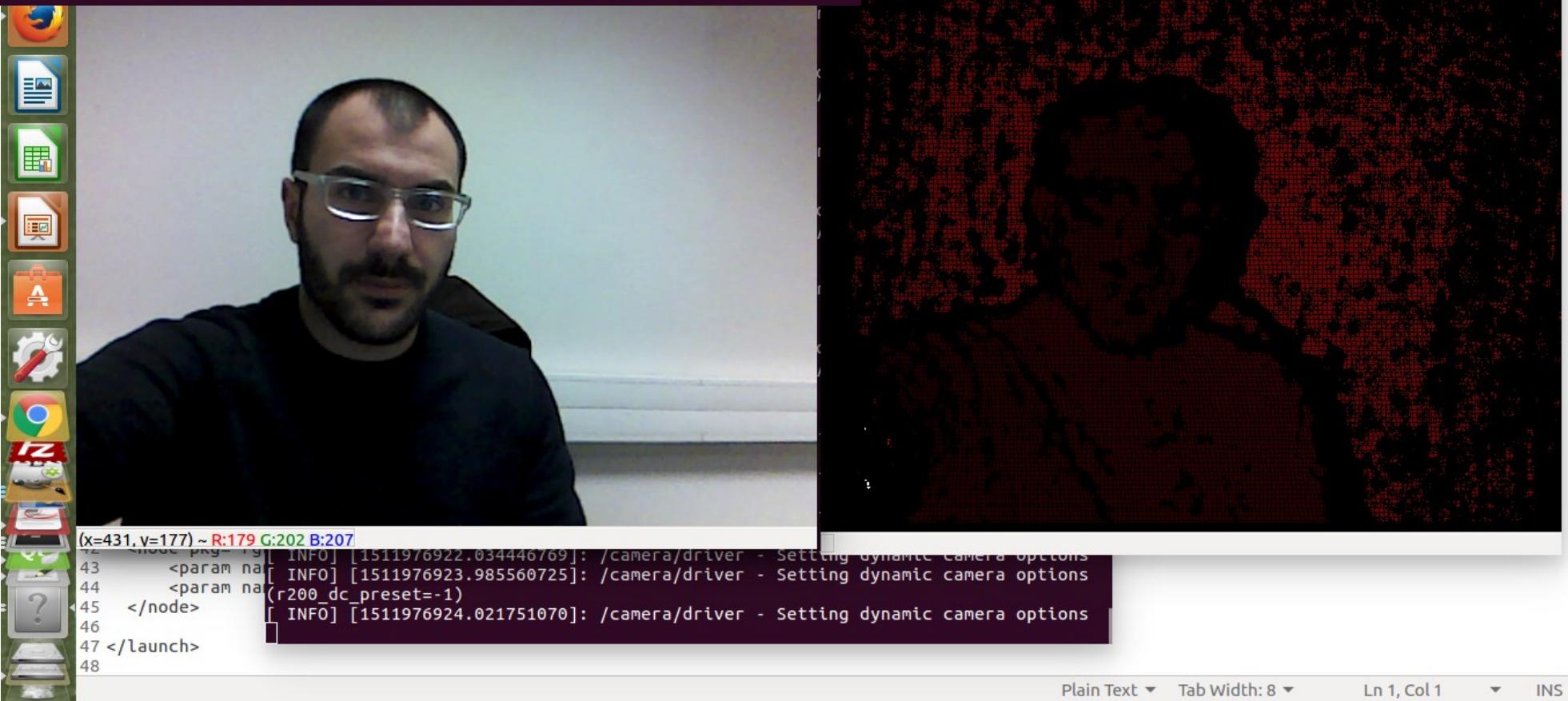
```
/opt/ros/kinetic/share/realsense_camera/launch/r200_nodelet_default.launch http://loc...  
[ INFO] [1511975468.786957868]: Initializing nodelet with 4 worker threads.  
[ INFO] [1511975469.042562486]: /camera	driver - Detected the following camera:  
- Serial No: 2511001026, USB Port ID: 4-1, Name:  
Intel RealSense R200, Camera FW: 1.0.71.06  
[ WARN] [1511975469.042746704]: /camera	driver - Detected unvalidated firmware:  
- 2511001026's current camera firmware is 1.0.71  
.06, Validated camera firmware is 1.0.72.06  
[ INFO] [1511975469.042882182]: /camera	driver - Connecting to camera with Seria  
l No: 2511001026, USB Port ID: 4-1  
[ INFO] [1511975469.618409936]: /camera	driver - Setting static camera options  
[ INFO] [1511975469.625166570]: /camera	driver - Enabling Depth in manual mode  
[ INFO] [1511975469.625406256]: /camera	driver - Enabling Color in manual mode  
[ INFO] [1511975469.625718834]: /camera	driver - Starting camera  
[ INFO] [1511975469.643368625]: /camera	driver - Publishing camera transforms (/  
tf_static)  
[ INFO] [1511975469.643552017]: /camera	driver - Setting dynamic camera options  
(r200_dc_preset=5)  
[ INFO] [1511975470.272138956]: /camera	driver - Initializing Depth Control Pres  
et to 5  
[ INFO] [1511975472.369285536]: /camera	driver - Setting dynamic camera options
```

sensore reale



rosrun (riproviamo)

```
bloisi@bloisi-U36SG:~/catkin_ws$ rosrun realsense_r200_viewer realsense_r200_vie  
wer  
Subscriptions:  
- RGB topic: camera/rgb/image_rect_color  
- Depth topic: camera/depth_registered/sw_registered/image_rect_raw  
init done
```



rosbag

Scarichiamo la ROS bag da

<http://profs.scienze.univr.it/~bloisi/corsi/lezionicf/office-r200.bag>

Si tratta di un file con una grandezza maggiore di 900MB

rosbag info

```
$ rosbag info office-r200.bag
```

```
bloisi@bloisi-U36SG:~/bagfiles$ ls
face.bag  office-r200.bag  TB3_WAFFLE_SLAM.bag
bloisi@bloisi-U36SG:~/bagfiles$ rosbag info office-r200.bag
path:      office-r200.bag
version:   2.0
duration:  59.8s
start:    Jan 25 2018 13:34:32.33 (1516883672.33)
end:     Jan 25 2018 13:35:32.16 (1516883732.16)
size:    921.0 MB
messages: 7172
compression: none [922/922 chunks]
types:    sensor_msgs/CameraInfo [c9a58c1b0b154e0e6da7578cb991d214]
          sensor_msgs/Image   [060021388200f6f0f447d0fc9c64743]
topics:   /camera/depth_registered/sw_registered/camera_info    1792 msgs : sensor_msgs/CameraInfo
          /camera/depth_registered/sw_registered/image_rect      1792 msgs : sensor_msgs/Image
          /camera/rgb/camera_info                           1796 msgs : sensor_msgs/CameraInfo
          /camera/rgb/image_rect_color                   1792 msgs : sensor_msgs/Image
bloisi@bloisi-U36SG:~/bagfiles$
```

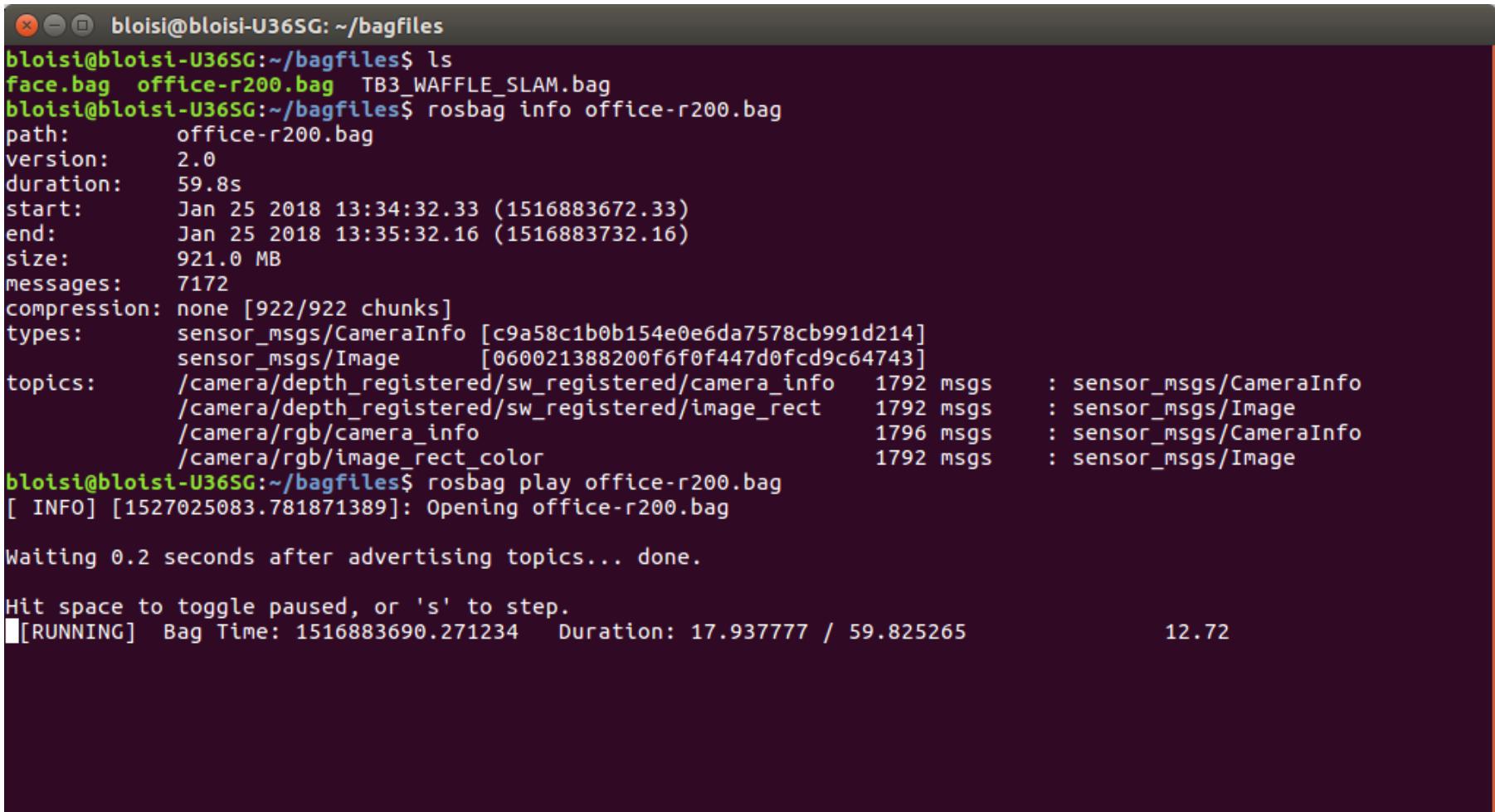
rosbag play

```
$ rosbag play office-r200.bag
```

```
bloisi@bloisi-U36SG: ~/bagfiles
bloisi@bloisi-U36SG:~/bagfiles$ ls
face.bag  office-r200.bag  TB3_WAFFLE_SLAM.bag
bloisi@bloisi-U36SG:~/bagfiles$ rosbag info office-r200.bag
path:      office-r200.bag
version:   2.0
duration:  59.8s
start:    Jan 25 2018 13:34:32.33 (1516883672.33)
end:     Jan 25 2018 13:35:32.16 (1516883732.16)
size:    921.0 MB
messages: 7172
compression: none [922/922 chunks]
types:    sensor_msgs/CameraInfo [c9a58c1b0b154e0e6da7578cb991d214]
          sensor_msgs/Image  [060021388200f6f0f447d0fc9c64743]
topics:   /camera/depth_registered/sw_registered/camera_info    1792 msgs : sensor_msgs/CameraInfo
          /camera/depth_registered/sw_registered/image_rect    1792 msgs : sensor_msgs/Image
          /camera/rgb/camera_info                           1796 msgs : sensor_msgs/CameraInfo
          /camera/rgb/image_rect_color                     1792 msgs : sensor_msgs/Image
bloisi@bloisi-U36SG:~/bagfiles$ rosbag play office-r200.bag
```

rosbag play - esecuzione

```
$ rosbag play office-r200.bag
```



The screenshot shows a terminal window with a dark background and light-colored text. The title bar of the terminal window reads "bloisi@bloisi-U36SG: ~/bagfiles". The terminal output is as follows:

```
bloisi@bloisi-U36SG:~/bagfiles$ ls
face.bag  office-r200.bag  TB3_WAFFLE_SLAM.bag
bloisi@bloisi-U36SG:~/bagfiles$ rosbag info office-r200.bag
path:      office-r200.bag
version:   2.0
duration:  59.8s
start:    Jan 25 2018 13:34:32.33 (1516883672.33)
end:     Jan 25 2018 13:35:32.16 (1516883732.16)
size:    921.0 MB
messages: 7172
compression: none [922/922 chunks]
types:    sensor_msgs/CameraInfo [c9a58c1b0b154e0e6da7578cb991d214]
          sensor_msgs/Image  [060021388200f6f0f447d0fc9c64743]
topics:   /camera/depth_registered/sw_registered/camera_info    1792 msgs   : sensor_msgs/CameraInfo
          /camera/depth_registered/sw_registered/image_rect    1792 msgs   : sensor_msgs/Image
          /camera/rgb/camera_info                      1796 msgs   : sensor_msgs/CameraInfo
          /camera/rgb/image_rect_color                 1792 msgs   : sensor_msgs/Image
bloisi@bloisi-U36SG:~/bagfiles$ rosbag play office-r200.bag
[ INFO] [1527025083.781871389]: Opening office-r200.bag
Waiting 0.2 seconds after advertising topics... done.

Hit space to toggle paused, or 's' to step.
[RUNNING] Bag Time: 1516883690.271234 Duration: 17.937777 / 59.825265           12.72
```

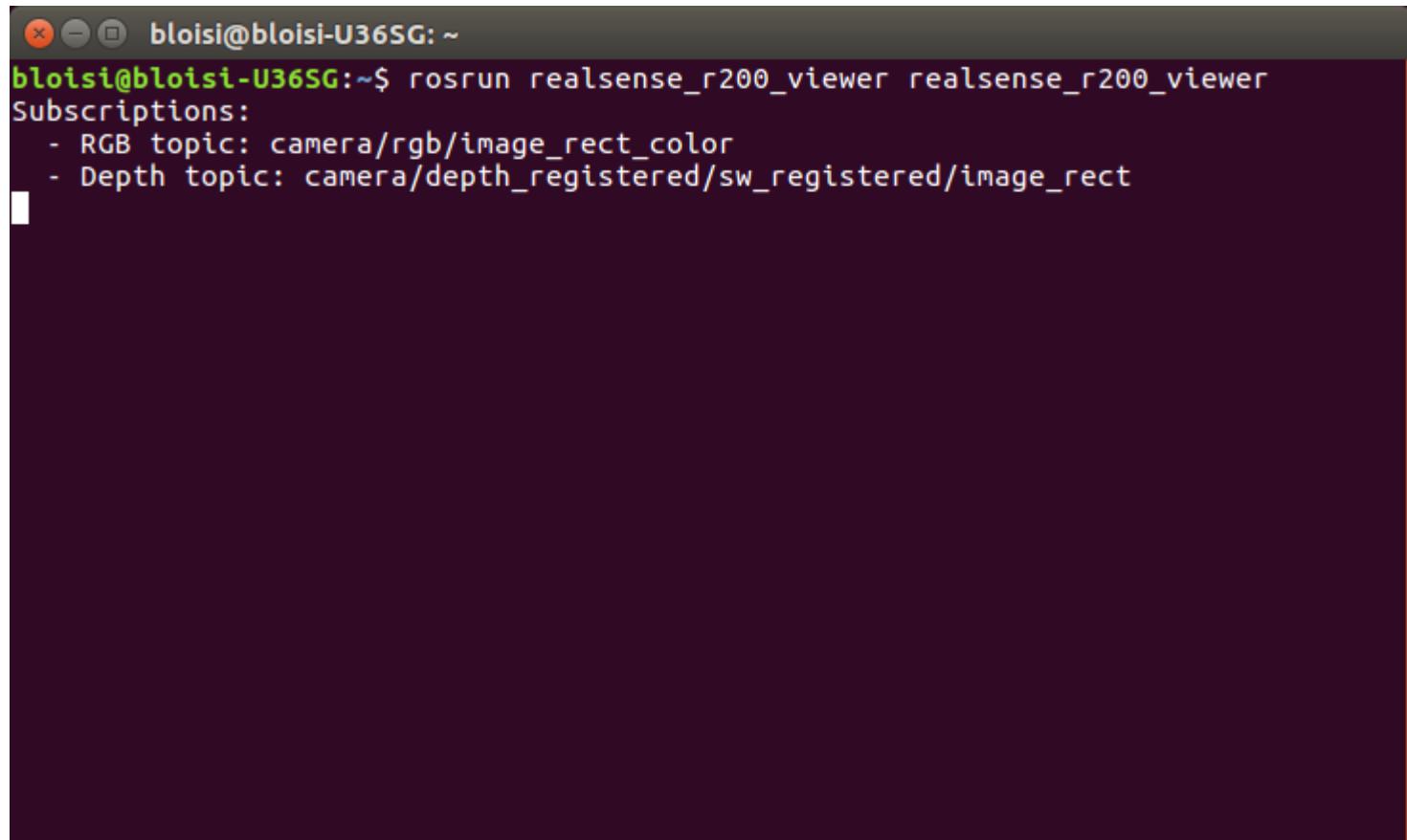
rosrun con bag file

```
$ rosrun realsense_r200_viewer real_sense_r200_viewer
```



rosrun con bag file - esecuzione

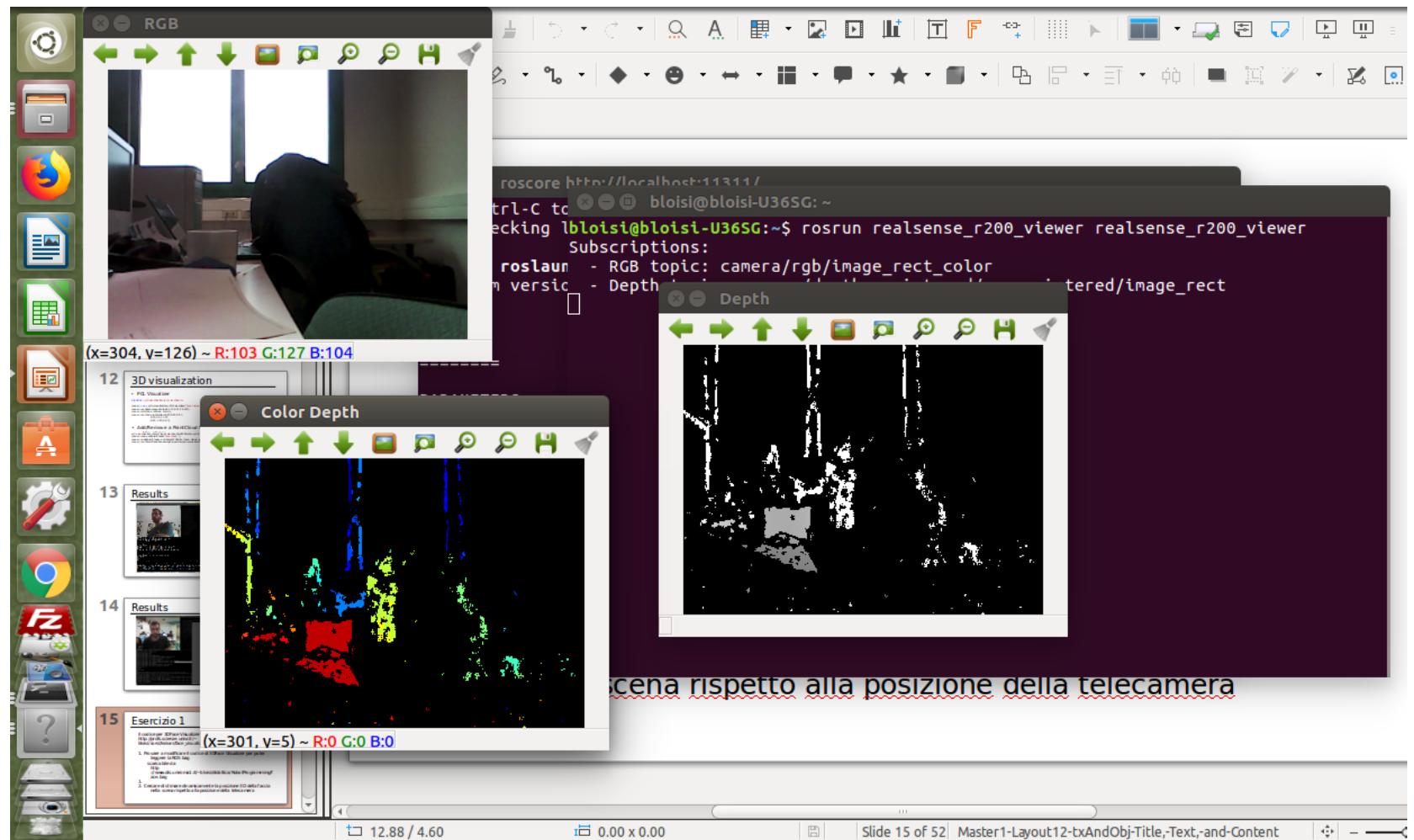
```
$ rosrun realsense_r200_viewer real_sense_r200_viewer
```

A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains the command "rosrun realsense_r200_viewer real_sense_r200_viewer" followed by the output "Subscriptions:" and two listed topics: "RGB topic: camera/rgb/image_rect_color" and "Depth topic: camera/depth_registered/sw_registered/image_rect".

```
bloisi@bloisi-U36SG:~$ rosrun realsense_r200_viewer real_sense_r200_viewer
Subscriptions:
- RGB topic: camera/rgb/image_rect_color
- Depth topic: camera/depth_registered/sw_registered/image_rect
```

rosrun con bag file - esecuzione

```
$ rosrun realsense_r200_viewer real_sense_r200_viewer
```



Esercizio

Scrivere un nodo ROS in grado di

1. leggere lo stream video proveniente dalla webcam integrata nel pc (o da una webcam esterna)
2. mostrare a video le immagini usando OpenCV

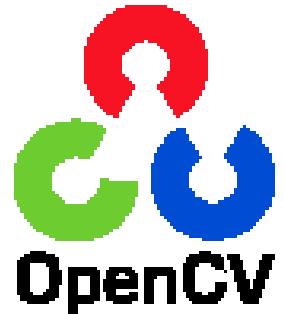
Pubblicare il codice usando un repository Git

3D Face Visualization

special thanks to
Roberto Capobianco and
Jacopo Serafin

Tools

- Microsoft Kinect or Asus Xtion
- OpenCV (Open Computer Vision)
- PCL (Point Cloud Library)
- ROS (Robot Operating System)

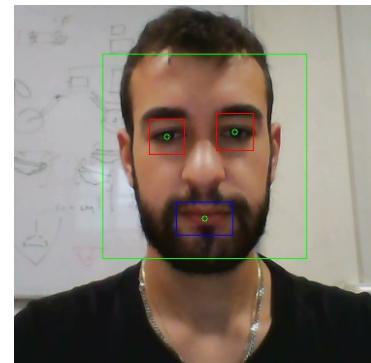


2D + Depth + 3D

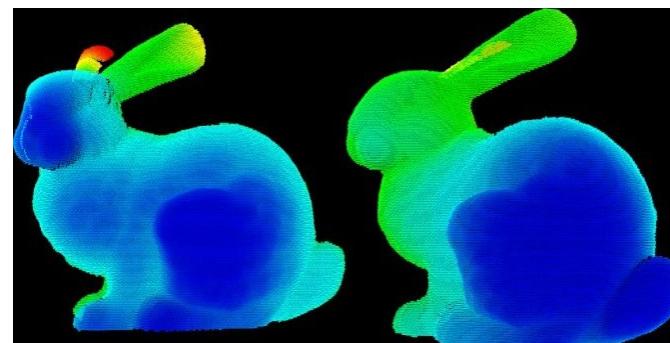
- Data Acquisition
(ROS + Kinect)
- Face Detection
(OpenCV)
- 3D Visualization
(PCL)



← Depth image



← RGB image



← 3D model

ROS topics

- **Kinect topic subscription**

Receive messages published by the Kinect node

Content of messages: Depth and RGB Images

Depth registered topic: one-by-one pixel

correspondence between Depth and RGB Images

- **Topic synchronization**

Required for processing pairs of Depth and RGB

Images close in terms of publishing time

ROS Callbacks

- **Callback function:**

- Binded to one or more (synchronized) topics

- Executed on a secondary thread whenever a new message is received

```
void callback(const ImageConstPtr& depthImage_) {  
    ...  
}  
  
void synchronized_callback(const ImageConstPtr& depthImage_,  
                         const ImageConstPtr& rgbImage_) {  
    ...  
}
```

Acquisizione dei dati

- Kinect topics
 - "/camera/depth_registered/image_rect_raw"
 - "/camera/rgb/image_rect"
- Topic subscription, synchronization and callback registration

```
#include <message_filters/subscriber.h>
#include <message_filters/synchronizer.h>
#include <message_filters/sync_policies/approximate_time.h>

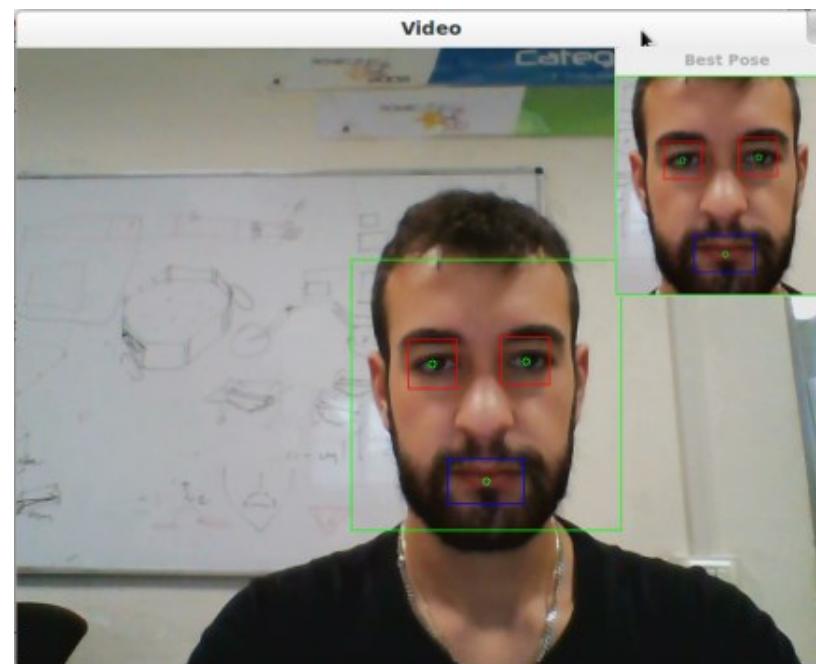
ros::NodeHandle nh;
message_filters::Subscriber<Image> depth_sub(nh, "topic1", 1);
message_filters::Subscriber<Image> rgb_sub(nh, "topic2", 1);

typedef sync_policies::ApproximateTime<Image, Image> syncPolicy;
Synchronizer<syncPolicy> sync(syncPolicy(10), depth_sub, rgb_sub);
sync.registerCallback(boost::bind(&callback, _1, _2));
```

Face Detection Demo

- Face detection on the whole image, both for frontal and profile faces

The face which is selected among the alternatives is the most visible one or, more formally, the face with the biggest area



Face Detection Demo

- Haar-like cascade declaration

```
#include <opencv2/opencv.hpp>
#include <opencv2/imgproc/imgproc.hpp>

cv::CascadeClassifier frontal_face_cascade;
cv::CascadeClassifier profile_face_cascade;

if(!frontal_face_cascade.load(frontalFaceCascadeFilename) ||
   !profile_face_cascade.load(profileFaceCascadeFilename)) {
    std::cerr << "Error while loading HAAR cascades." << std::endl;
    return -1;
}
```

- Search the feature in the RGB image

```
frontal_face_cascade.detectMultiScale(grayImage, frontal_face_vector, 1.4, 4,
                                       0|CV_HAAR_SCALE_IMAGE, cv::Size(50, 50));
profile_face_cascade.detectMultiScale(grayImage, profile_face_vector, 1.4, 4,
                                       0|CV_HAAR_SCALE_IMAGE, cv::Size(50, 50));
```

3D visualization

PCL Visualizer is PCL's full-featured visualization class

- PointCloud visualization with RGB information
- Normal displaying
- Shape drawing
- Multiple viewports

2D to 3D

Depth point to 3D Cartesian point

$$\mathbf{p} = \mathbf{K}^{-1} \cdot (u, v, 1)^T$$

```
pcl::PointCloud::Ptr face_cloud(new pcl::PointCloud);
float cx = 319.5f; //optical center x coordinate
float cy = 239.5f; //optical center y coordinate
float f = 525.0f; //focal length (the same for x and y)
pcl::PointXYZRGB point;
point.z = d / 1000.0f;
point.x = (imageWidth - cx) * point.z / f;
point.y = (imageHeight - cy) * point.z / f;
cv::Vec3b pixel = rgbImage.at<cv::Vec3b>(imageHeight, imageWidth);
point.r = pixel[2];
point.g = pixel[1];
point.b = pixel[0];
face_cloud->points.push_back(point);
```

3D visualization

- PCL Visualizer

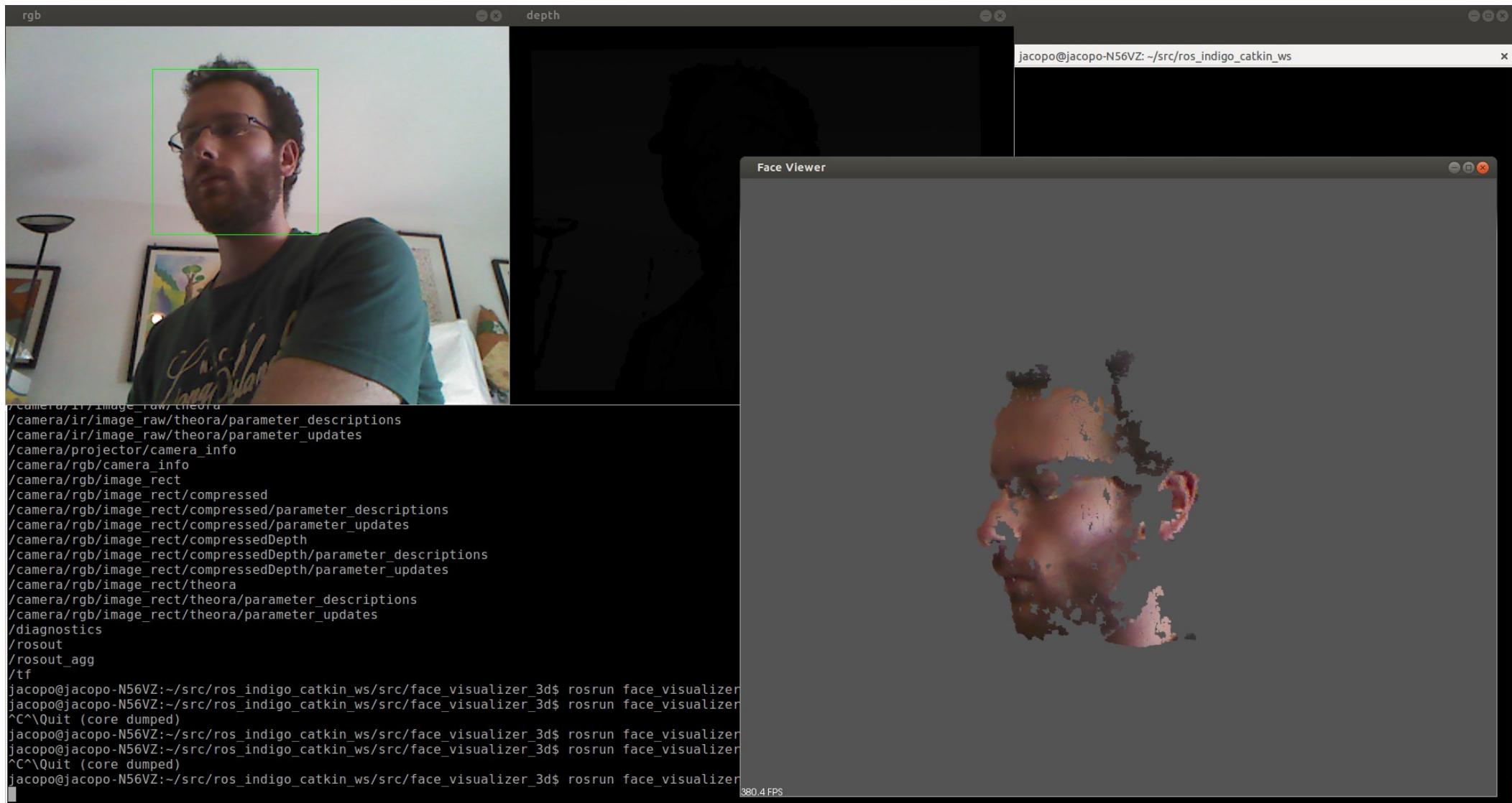
```
#include <pcl/visualization/pcl_visualizer.h>

viewer = new pcl::visualization::PCLVisualizer("Face Viewer");
viewer->setBackgroundColor(0.33f, 0.33f, 0.33f);
viewer->initCameraParameters();
viewer->setCameraPosition(0.0f, 0.0f, 0.0f,
                         0.0f, 0.0f, 1.0f,
                         0.0f, -1.0f, 0.0f);
```

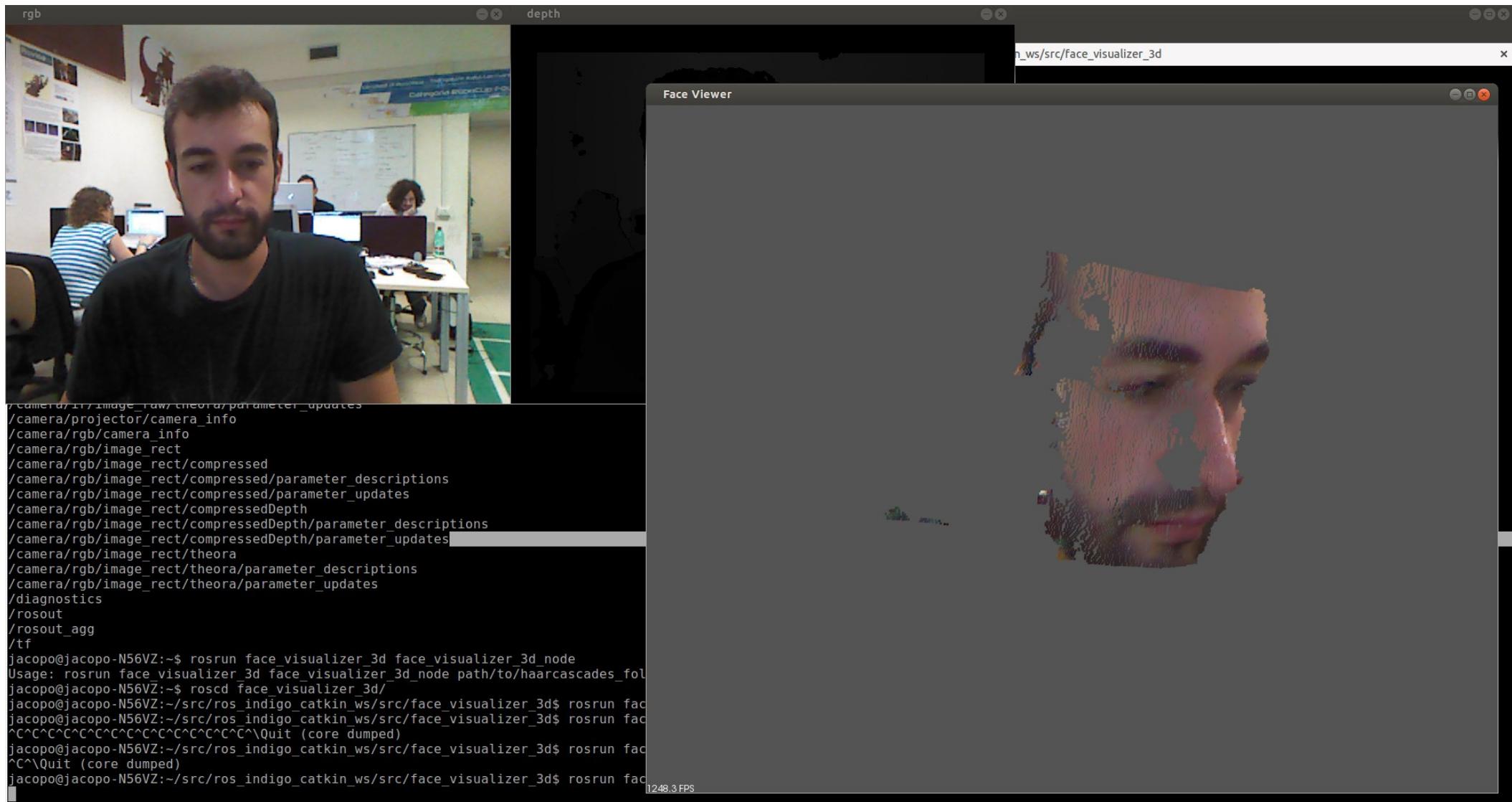
- Add/Remove a PointCloud to the Visualizer

```
pcl::visualization::PointCloudColorHandlerRGBField<pcl::PointXYZRGB> rgbHandler(face_cloud);
viewer->removePointCloud("face cloud");
viewer->addPointCloud<pcl::PointXYZRGB>(face_cloud, rgbHandler, "face cloud");
viewer->setPointCloudRenderingProperties(pcl::visualization::PCL_VISUALIZER_POINT_SIZE, 3, "face cloud");
```

Results



Results



Esercitazione

Il codice per 3DFace Visualizer è scaricabile da

http://profs.scienze.univr.it/~bloisi/corsi/lezioni/face_visualizer_3d.zip

1. Utilizzare 3DFace Visualizer per leggere la ROS bag
scaricabile da

<http://www.dis.uniroma1.it/~bloisi/didattica/RobotProgramming/face.bag>

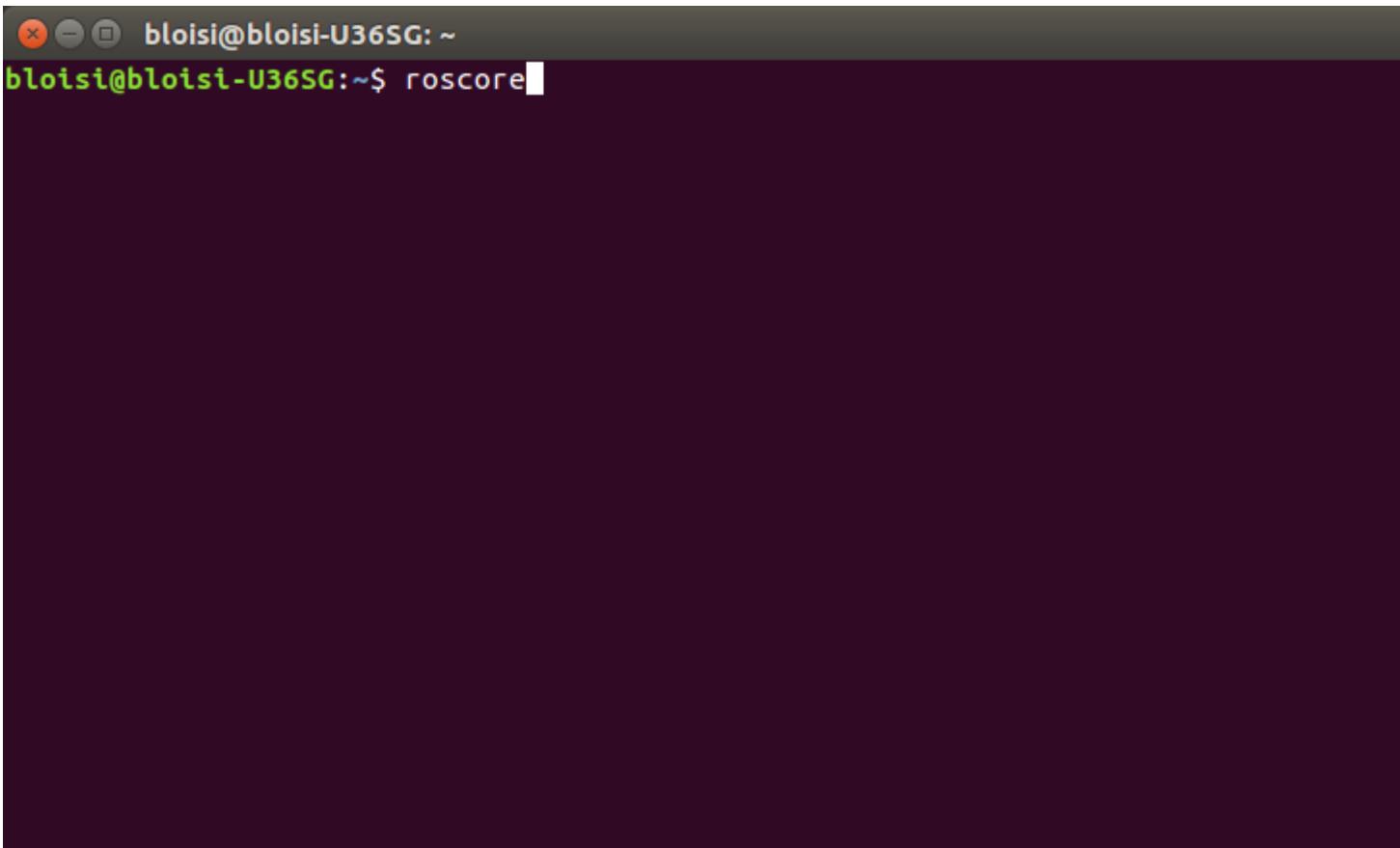
2. Visualizzare la nuvola di punti relativa al volto nelle immagini

rosbag info

Una volta terminato il download verifichiamo che i dati nella bag siano corretti

```
bloisi@bloisi-U36SG:~/bagfiles
bloisi@bloisi-U36SG:~/bagfiles$ rosbag info face.bag
path:      face.bag
version:   2.0
duration:  39.9s
start:    Oct 29 2014 15:01:16.62 (1414591276.62)
end:     Oct 29 2014 15:01:56.51 (1414591316.51)
size:    3.1 GB
messages: 10538
compression: none [1271/1271 chunks]
types:    diagnostic_msgs/DiagnosticArray      [60810da900de1dd6ddd437c3503511da]
          dynamic_reconfigure/Config           [958f16a05573709014982821e6822580]
          dynamic_reconfigure/ConfigDescription [757ce9d44ba8ddd801bb30bc456f946f]
          rosgraph_msgs/Log                  [acffd30cd6b6de30f120938c17c593fb]
          sensor_msgs/CameraInfo            [c9a58c1b0b154e0e6da7578cb991d214]
          sensor_msgs/CompressedImage       [8f7a12909da2c9d3332d540a0977563f]
          sensor_msgs/Image                 [060021388200f6f0f447d0fcfd9c64743]
          sensor_msgs/PointCloud2           [1158d486dd51d683ce2f1be655c3c181]
          stereo_msgs/DisparityImage        [04a177815f75271039fa21f16acad8c9]
          tf2_msgs/TFMessage                [94810edda583a504dfda3829e70d7eec]
          theora_image_transport/Packet     [33ac4e14a7cff32e7e0d65f18bb410f3]
topics:   /camera/debayer/parameter_descriptions      1
          msg   : dynamic_reconfigure/ConfigDescription
                  /camera/depth/image/compressed/parameter_descriptions
          msg   : dynamic_reconfigure/Config
                  /camera/depth/image/compressedDepth/parameter_descriptions
          msg   : dynamic_reconfigure/ConfigDescription
                  /camera/depth/image/compressedDepth/parameter_updates
          msg   : dynamic_reconfigure/Config
                  /camera/depth/image/compressedDepth/parameter_descriptions
          msg   : dynamic_reconfigure/ConfigDescription
                  /camera/depth/image/compressedDepth/parameter_updates
```

roscore



A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains the command "bloisi@bloisi-U36SG:~\$ roscore" which has been partially typed into the input field. The rest of the terminal window is blank.

roscore - esecuzione

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

started roslaunch server http://localhost:43877/
ros_comm version 1.12.13

SUMMARY
=====

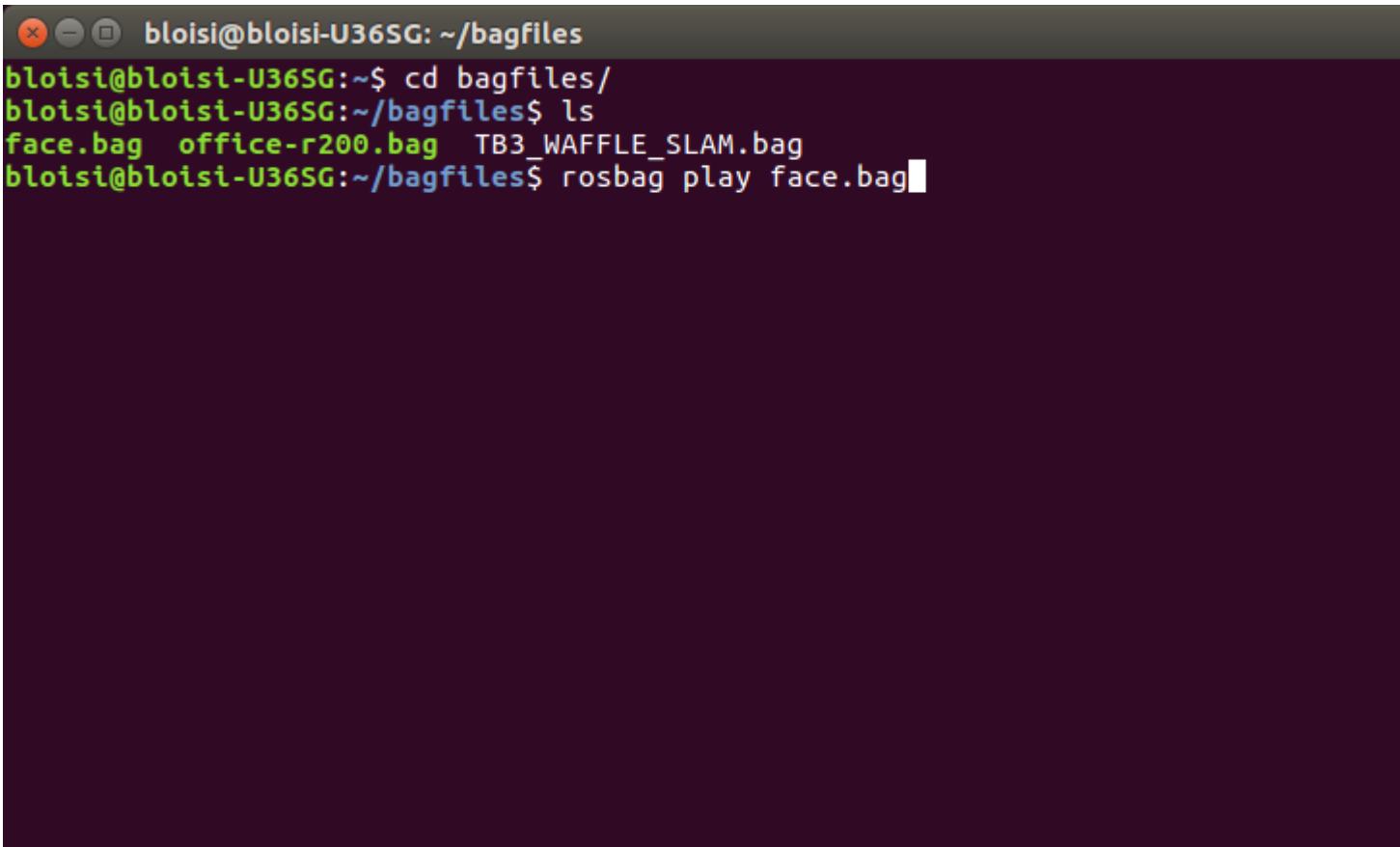
PARAMETERS
* /rostdistro: kinetic
* /rosversion: 1.12.13

NODES

auto-starting new master
process[master]: started with pid [5033]
ROS_MASTER_URI=http://localhost:11311/

setting /run_id to 42b80570-5e06-11e8-aff0-57e3c576faa3
process[rosout-1]: started with pid [5046]
started core service [/rosout]
```

rosbag play

A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~/bagfiles". The terminal displays a command-line session:

```
bloisi@bloisi-U36SG:~/bagfiles
bloisi@bloisi-U36SG:~$ cd bagfiles/
bloisi@bloisi-U36SG:~/bagfiles$ ls
face.bag  office-r200.bag  TB3_WAFFLE_SLAM.bag
bloisi@bloisi-U36SG:~/bagfiles$ rosbag play face.bag
```

The terminal has a dark background and light-colored text. The cursor is visible at the end of the command "face.bag".

bloisi@bloisi-U36SG:~/bagfiles
bloisi@bloisi-U36SG:~\$ cd bagfiles/
bloisi@bloisi-U36SG:~/bagfiles\$ ls
face.bag office-r200.bag TB3_WAFFLE_SLAM.bag
bloisi@bloisi-U36SG:~/bagfiles\$ rosbag play face.bag

rosbag play - esecuzione

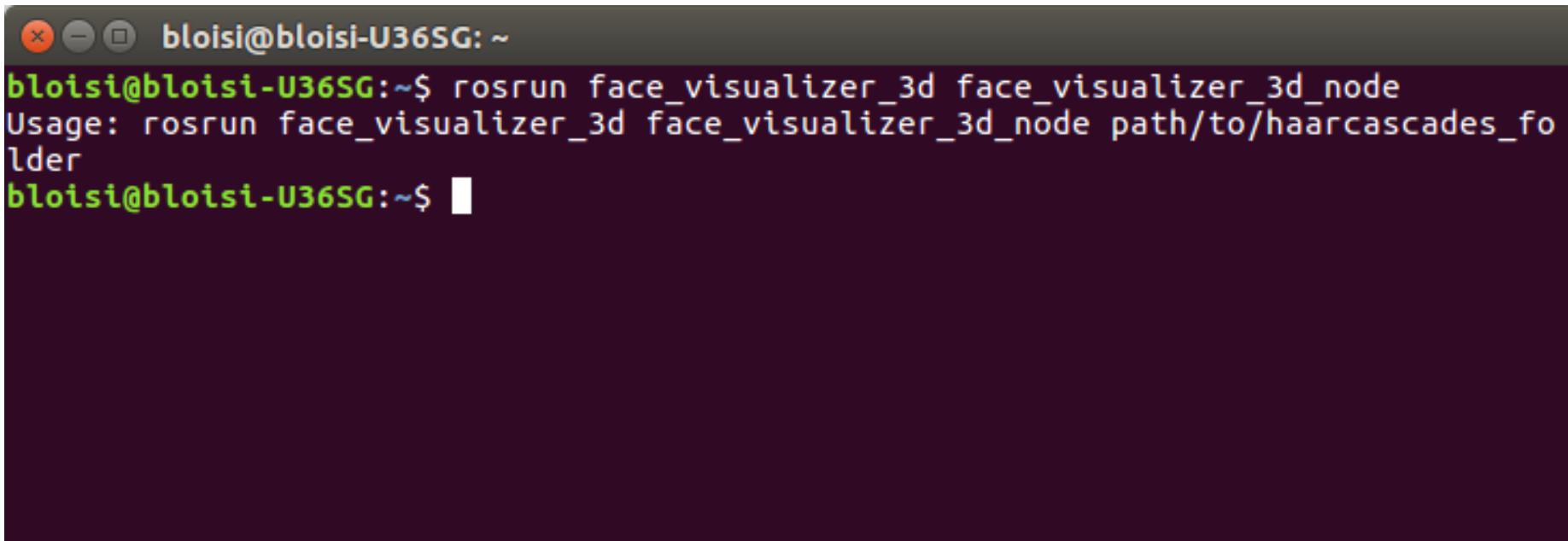
```
bloisi@bloisi-U36SG: ~/bagfiles
[RUNNING] Bag Time: 1414591277.276300 Duration: 0.660924 / 39.898938
[RUNNING] Bag Time: 1414591277.278651 Duration: 0.663275 / 39.898938
[RUNNING] Bag Time: 1414591277.280668 Duration: 0.665293 / 39.898938
[RUNNING] Bag Time: 1414591277.289902 Duration: 0.674526 / 39.898938
[RUNNING] Bag Time: 1414591277.303108 Duration: 0.687732 / 39.898938
[RUNNING] Bag Time: 1414591277.306389 Duration: 0.691013 / 39.898938
[RUNNING] Bag Time: 1414591277.307226 Duration: 0.691850 / 39.898938
[RUNNING] Bag Time: 1414591277.312841 Duration: 0.697465 / 39.898938
[RUNNING] Bag Time: 1414591277.313306 Duration: 0.697930 / 39.898938
[RUNNING] Bag Time: 1414591277.316171 Duration: 0.700795 / 39.898938
[RUNNING] Bag Time: 1414591277.331296 Duration: 0.715920 / 39.898938
[RUNNING] Bag Time: 1414591277.341717 Duration: 0.726342 / 39.898938
[RUNNING] Bag Time: 1414591277.343130 Duration: 0.727754 / 39.898938
[RUNNING] Bag Time: 1414591277.366968 Duration: 0.751592 / 39.898938
[RUNNING] Bag Time: 1414591277.378958 Duration: 0.763582 / 39.898938
[RUNNING] Bag Time: 1414591277.383366 Duration: 0.767990 / 39.898938
[RUNNING] Bag Time: 1414591277.385019 Duration: 0.769643 / 39.898938
[RUNNING] Bag Time: 1414591277.403539 Duration: 0.788164 / 39.898938
[RUNNING] Bag Time: 1414591277.547314 Duration: 0.931938 / 39.898938
[RUNNING] Bag Time: 1414591277.550498 Duration: 0.935122 / 39.898938
[RUNNING] Bag Time: 1414591277.553698 Duration: 0.938322 / 39.898938
[RUNNING] Bag Time: 1414591277.567971 Duration: 0.952595 / 39.898938
[RUNNING] Bag Time: 1414591277.594067 Duration: 0.978692 / 39.898938
```

rosrun

A screenshot of a terminal window from an Ubuntu desktop environment. The window title bar shows the user's name, "bloisi", followed by the host name, "bloisi-U36SG", and the directory, "~". The terminal itself is dark-themed. In the top-left corner of the terminal area, there is a small white icon containing a red circle with a white cross, likely a signal or error indicator. The command entered is "rosrun face_visualizer_3d face_visualizer_3d_node", which is displayed in green text, indicating it is a command. The rest of the terminal window is blank.

```
bloisi@bloisi-U36SG:~$ rosrun face_visualizer_3d face_visualizer_3d_node
```

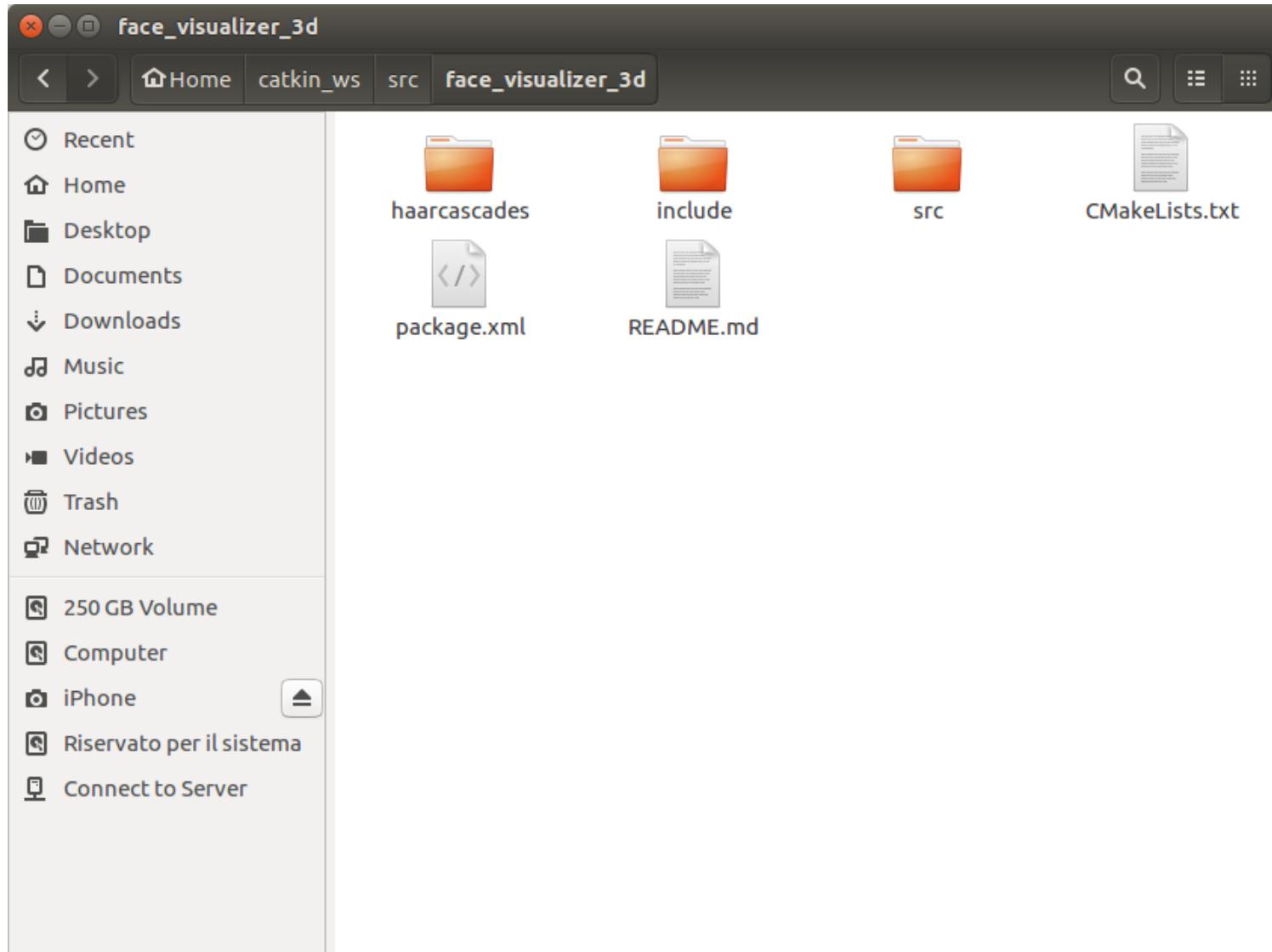
rosrun - esecuzione

A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains the following text:

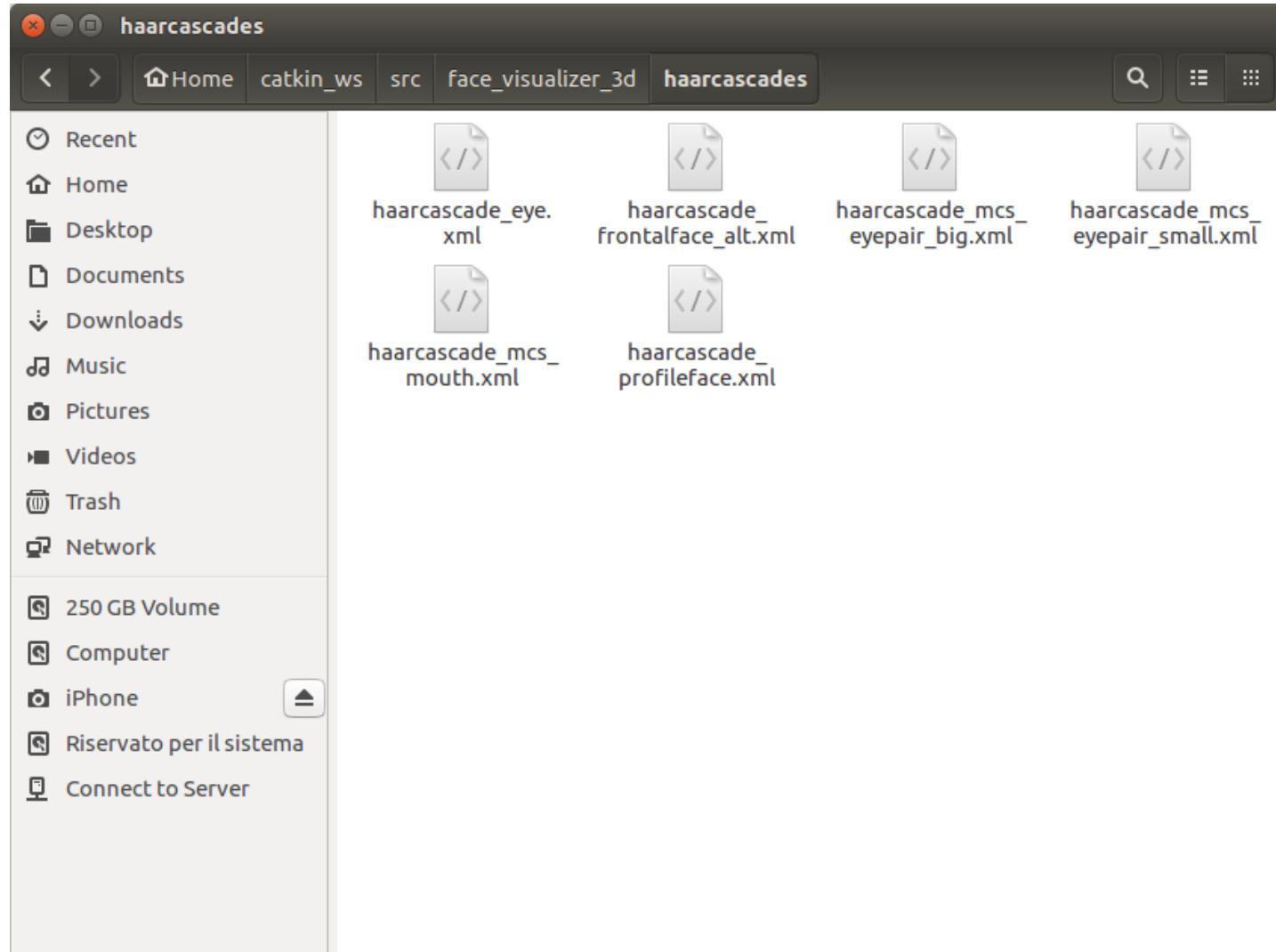
```
bloisi@bloisi-U36SG:~$ rosrun face_visualizer_3d face_visualizer_3d_node
Usage: rosrun face_visualizer_3d face_visualizer_3d_node path/to/haarcascades_
lder
bloisi@bloisi-U36SG:~$
```

manca il
classificatore!

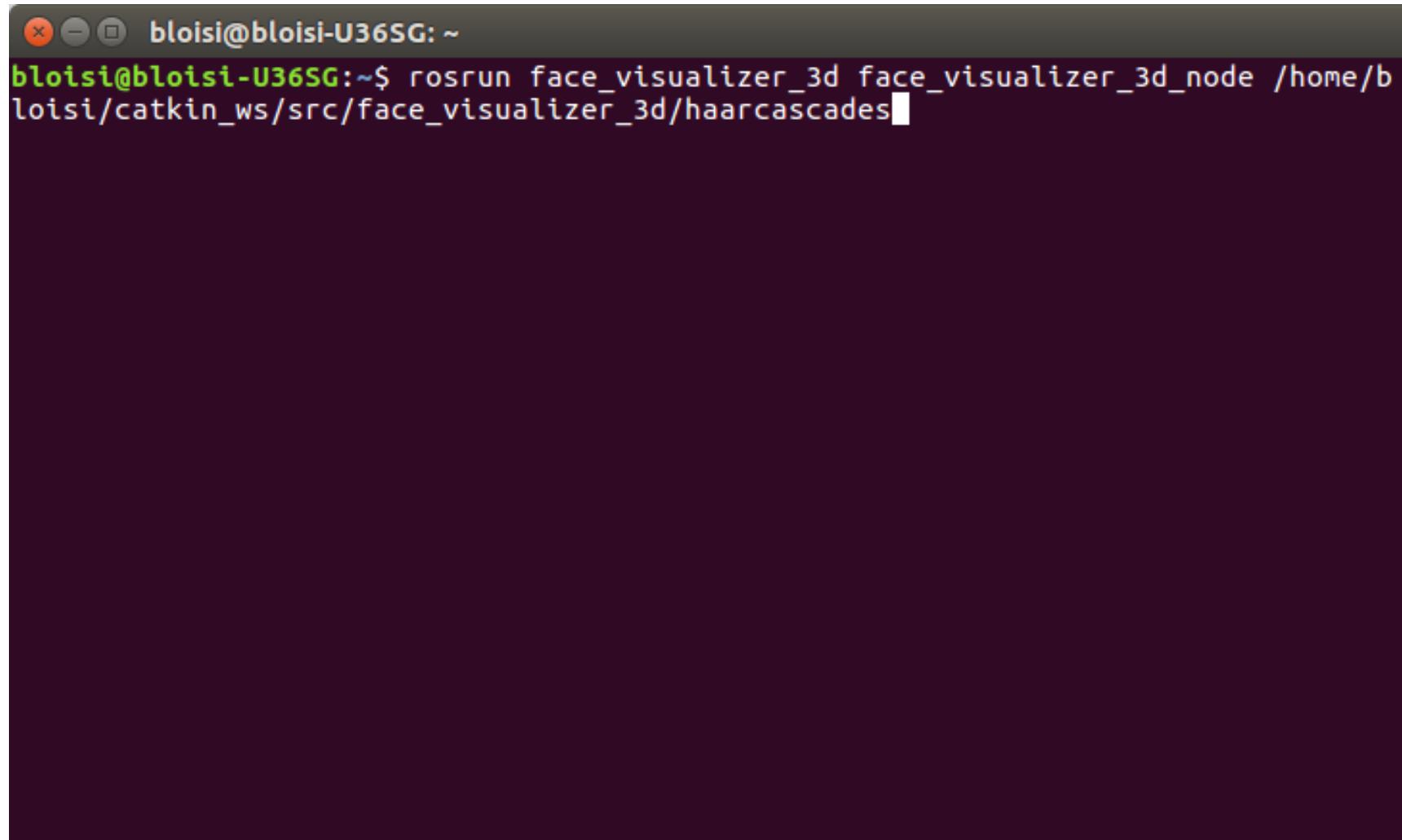
Classificatore



Classificatori forniti da OpenCV

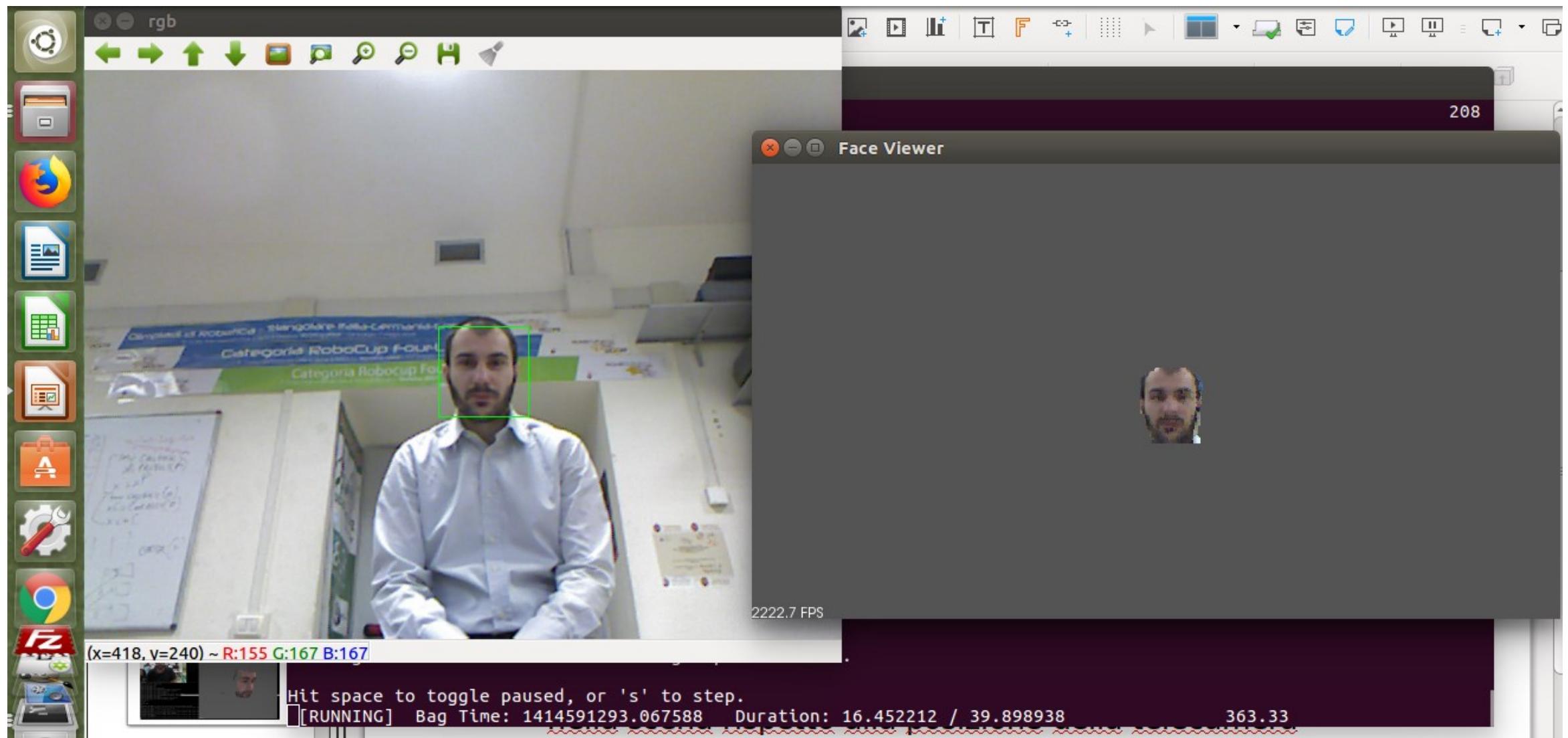


rosrun corretto con classificatore

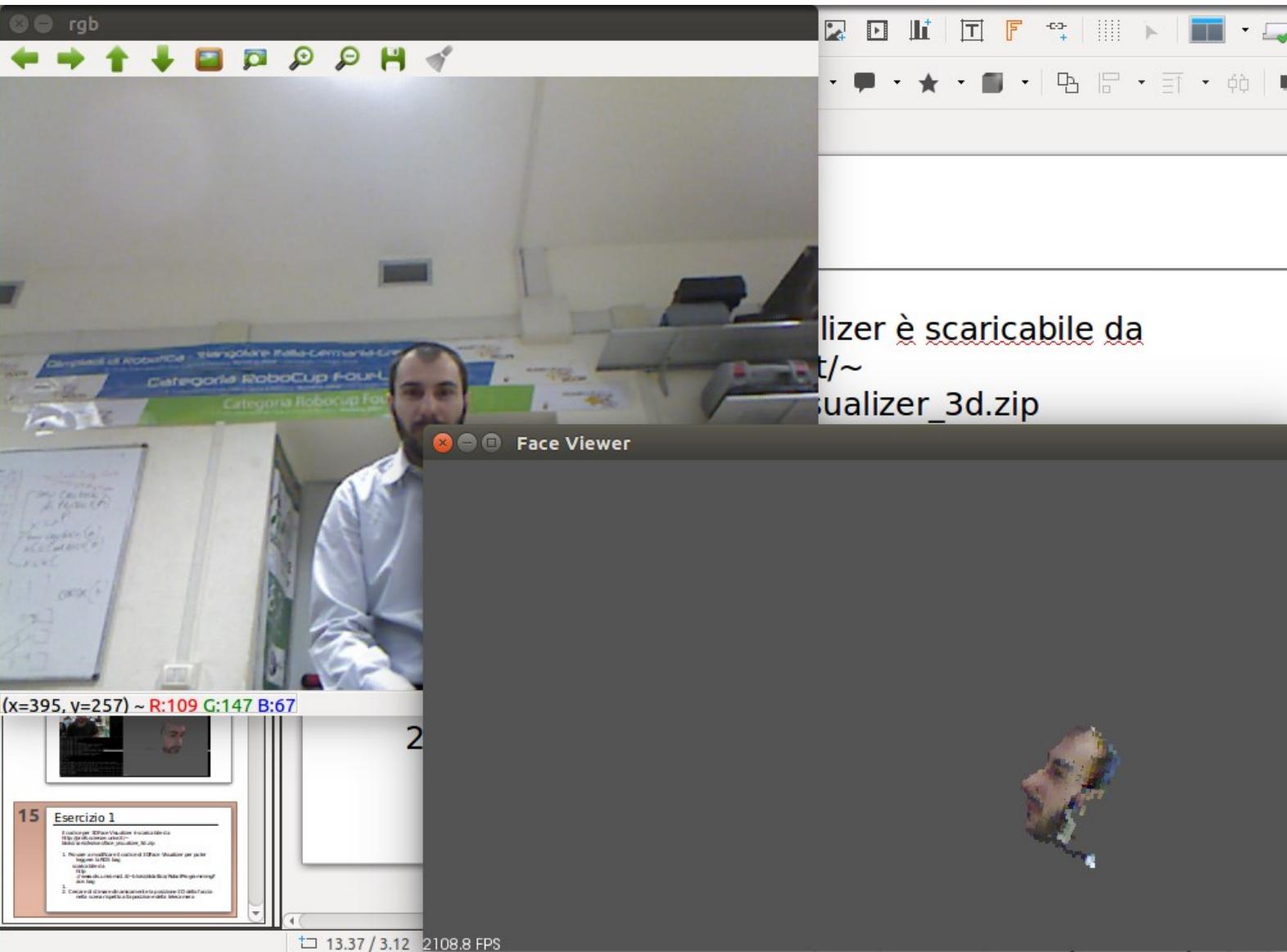
A screenshot of a terminal window titled "bloisi@bloisi-U36SG: ~". The window contains a single command: "bloisi@bloisi-U36SG:~\$ rosrun face_visualizer_3d face_visualizer_3d_node /home/bloisi/catkin_ws/src/face_visualizer_3d/haarcascades".

```
bloisi@bloisi-U36SG:~$ rosrun face_visualizer_3d face_visualizer_3d_node /home/bloisi/catkin_ws/src/face_visualizer_3d/haarcascades
```

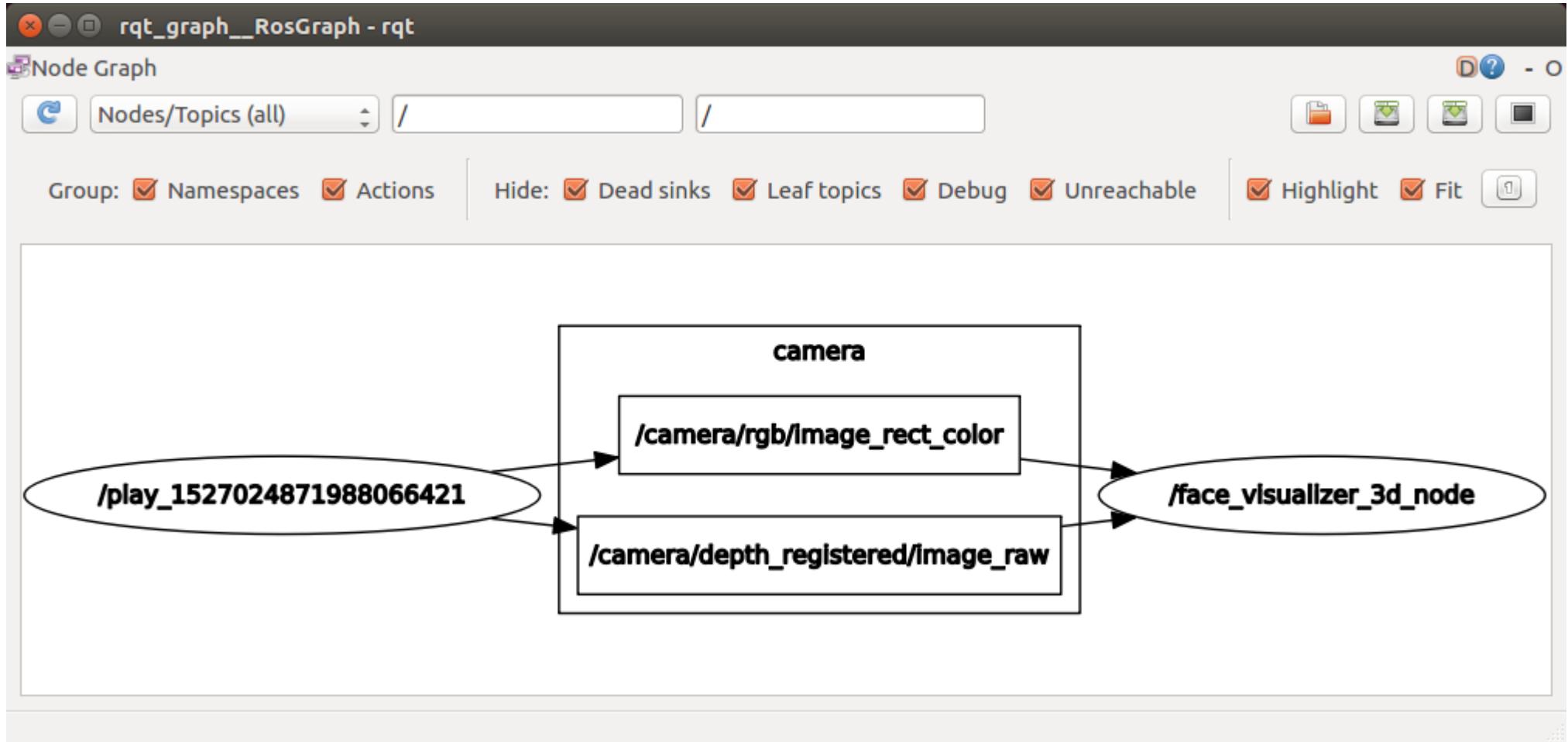
esecuzione



Visualizzazione 3D



rqt_graph



Esercizio

1. Provare a modificare il codice di 3DFace Visualizer per poter leggere la ROS bag scaricabile da
<http://www.dis.uniroma1.it/~bloisi/didattica/RobotProgramming/people.bag>
2. Cercare di estrarre oltre al volto anche il corpo delle persone nella bag



UNIVERSITÀ
di VERONA

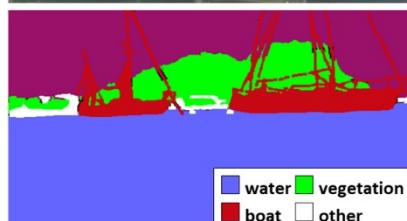
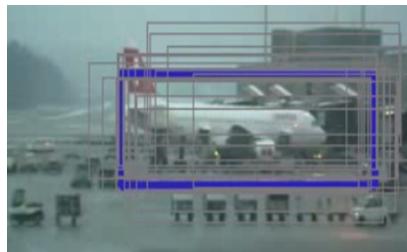
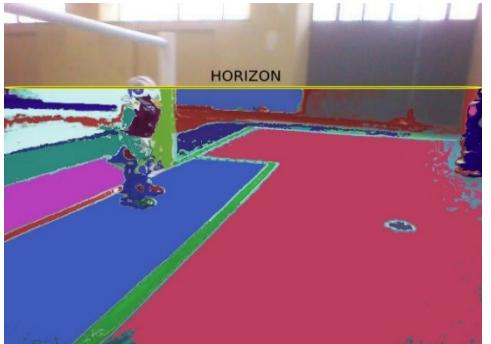
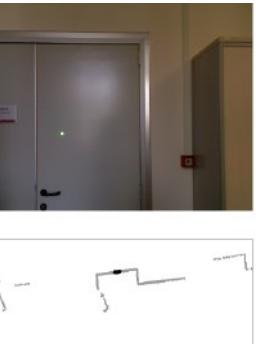
Dipartimento
di INFORMATICA

Corso di Laboratorio Ciberfisico
Modulo di Robot Programming with ROS

Esercitazione

3d face
visualization

Maggio 2018



Docente:
Domenico Daniele
Bloisi

■ water ■ vegetation
■ boat ■ other