



UNIVERSITÀ
di **VERONA**

Dipartimento
di **INFORMATICA**

Laurea magistrale in ingegneria e scienze informatiche

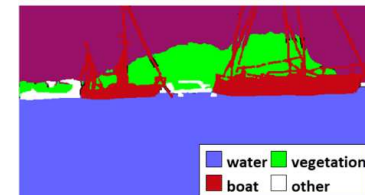
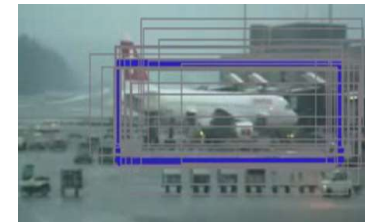
 **ROS** +
 **git**



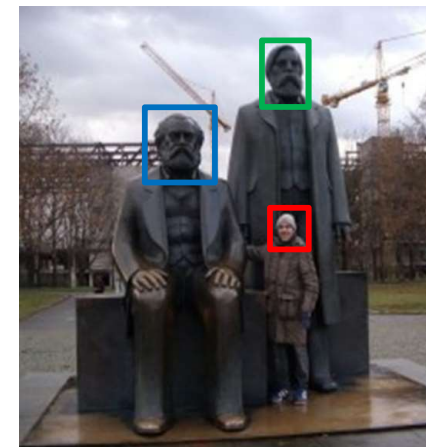
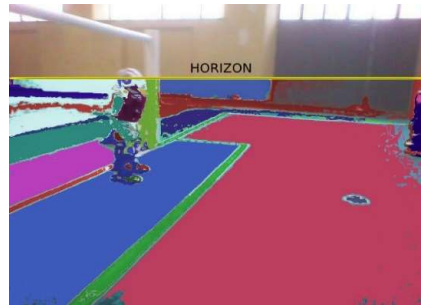
*Corso di Robotica
Parte di Laboratorio*

Docente:

Domenico Daniele Bloisi



Novembre 2017



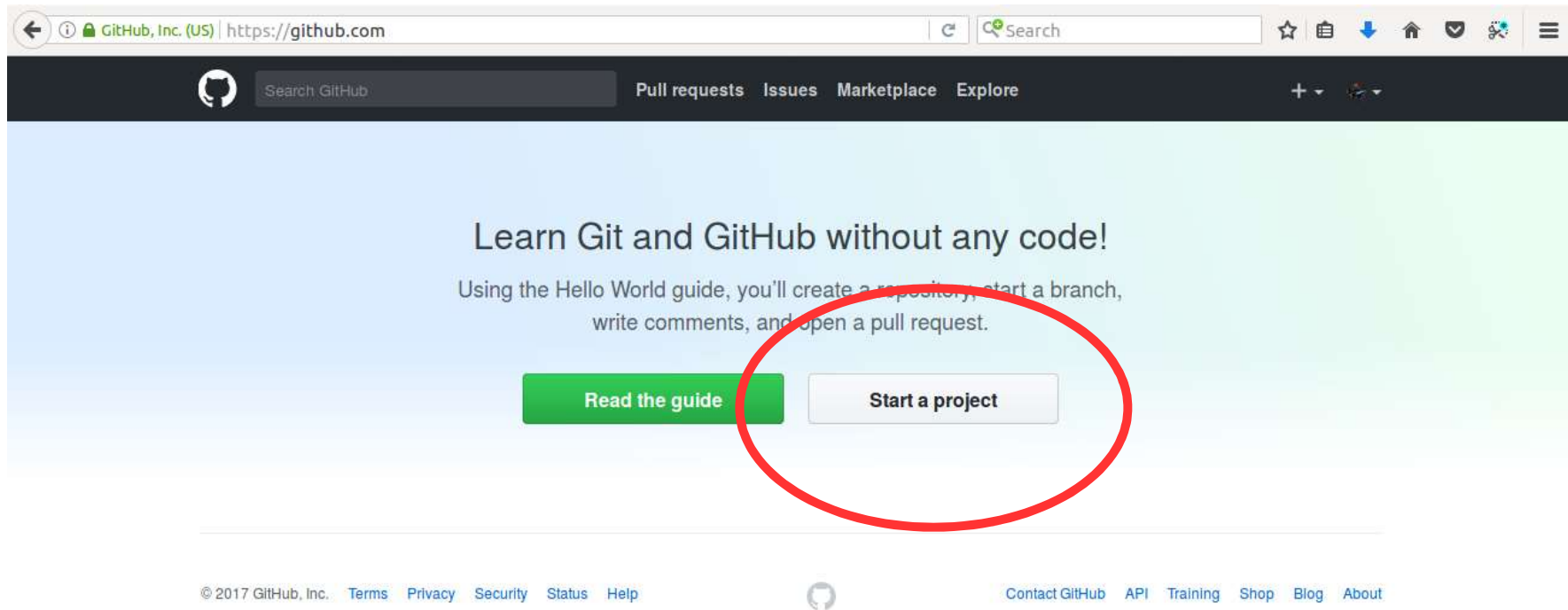
ROS + git

Esempio pratico

1. creare un nodo ROS
2. creare un repository git
3. condividere il nodo ROS
tramite il repository git
4. modificare il nodo ROS
usando git



Creare un repository git



Repository name

GitHub, Inc. (US) | https://github.com/new

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

labrobotica-labrobotica

realsense_r200_viewer

Great repository names are short and memorable. Need inspiration? How about [fuzzy-eureka](#).

Description (optional)

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



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.



Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add a README

Add a license: **None**

Create repository

Repository creato

The screenshot shows a GitHub repository page for 'labrobotica-bloisi/realSense_r200_viewer'. The repository name is circled in red. Below the repository name, the description 'ROS node for visualizing data coming from an Intel RealSense R200 device' is visible. The repository statistics show 2 commits, 1 branch, 0 releases, and 1 contributor. The 'Clone or download' button is highlighted in green. The README file is listed, and its content is circled in red, showing the title 'realSense_r200_viewer' and the same description as the repository page.

labrobotica-bloisi / **realSense_r200_viewer** Watch 0 Star 0 Fork 0

ROS node for visualizing data coming from an Intel RealSense R200 device [Edit](#)

[Add topics](#)

2 commits 1 branch 0 releases 1 contributor

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

labrobotica-bloisi Initial commit Latest commit c3aab1e an hour ago

[README.md](#) Initial commit an hour ago

[README.md](#)

realSense_r200_viewer

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

clone

The image shows a screenshot of a GitHub repository page for 'labrobotica-bloisi / realsense_r200_viewer'. The repository description is 'ROS node for visualizing data coming from an Intel RealSense R200 device'. The page shows 2 commits, 1 branch, 0 releases, and 1 contributor. A red circle highlights the 'Clone or download' button and its dropdown menu, which includes options for 'Clone with HTTPS', 'Use SSH', and 'Download ZIP'. The HTTPS URL is 'https://github.com/labrobotica-bloisi/realsense_r200_viewer'. The repository name 'realsense_r200_viewer' is displayed in a large font at the bottom of the page.

labrobotica-bloisi / realsense_r200_viewer

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

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

2 commits 1 branch 0 releases 1 contributor

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

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

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

```
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



Creating a ROS package by hand

The very first thing we'll do is add our [manifest](#) file.

The package.xml file allows tools like [rospack](#) to determine information about what your package depends upon.



<http://wiki.ros.org/ROS/Tutorials/Creating%20a%20Package%20by%20Hand>

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>

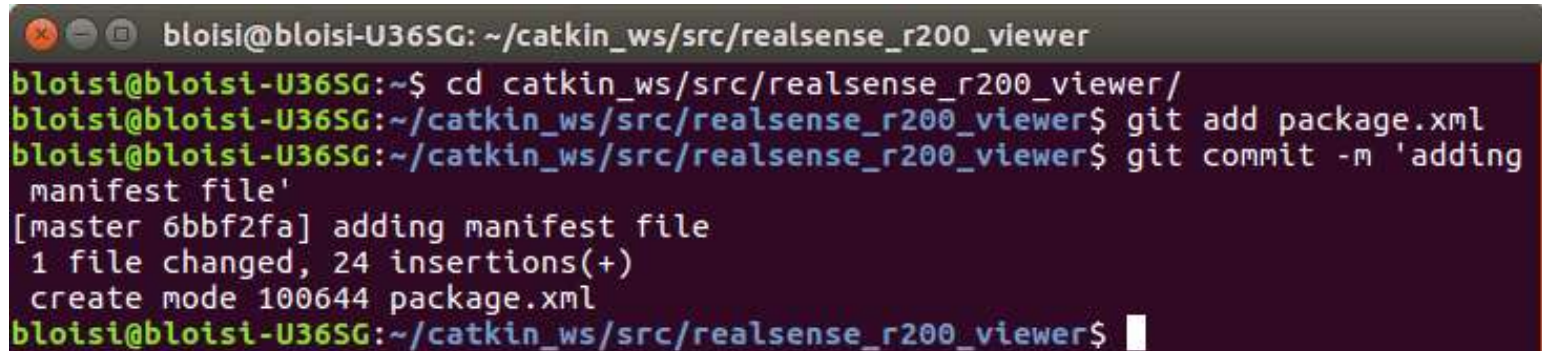
  <build_depend>cv_bridge</build_depend>
  <build_depend>image_transport</build_depend>
  <build_depend>roscpp</build_depend>
  <build_depend>rospy</build_depend>
  <build_depend>sensor_msgs</build_depend>
  <build_depend>std_msgs</build_depend>

  <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>
```

Adding package.xml

git add

git commit

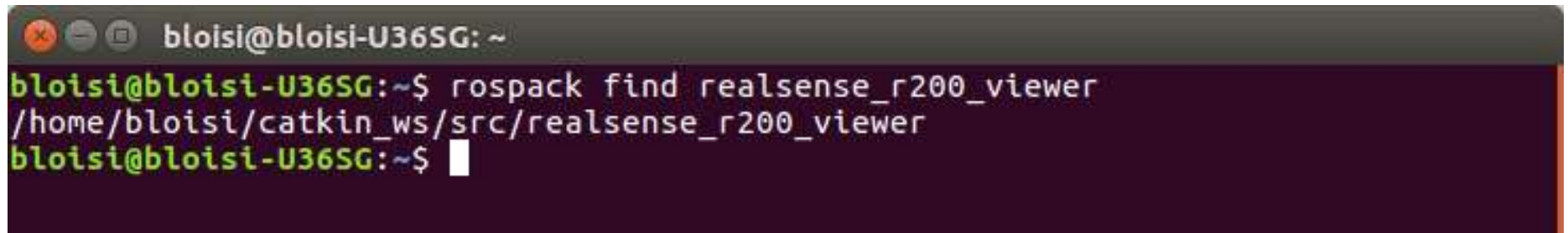


```
bloisi@bloisi-U36SG: ~/catkin_ws/src/realsense_r200_viewer
bloisi@bloisi-U36SG:~$ cd catkin_ws/src/realsense_r200_viewer/
bloisi@bloisi-U36SG:~/catkin_ws/src/realsense_r200_viewer$ git add package.xml
bloisi@bloisi-U36SG:~/catkin_ws/src/realsense_r200_viewer$ git commit -m 'adding
manifest file'
[master 6bbf2fa] adding manifest file
1 file changed, 24 insertions(+)
create mode 100644 package.xml
bloisi@bloisi-U36SG:~/catkin_ws/src/realsense_r200_viewer$
```

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 terminal window with a dark background and light text. The window title is 'bloisi@bloisi-U36SG: ~'. The prompt is 'bloisi@bloisi-U36SG:~\$'. The command 'rospack find realsense_r200_viewer' is entered. The output is '/home/bloisi/catkin_ws/src/realsense_r200_viewer'. The prompt is repeated as 'bloisi@bloisi-U36SG:~\$' with a cursor.

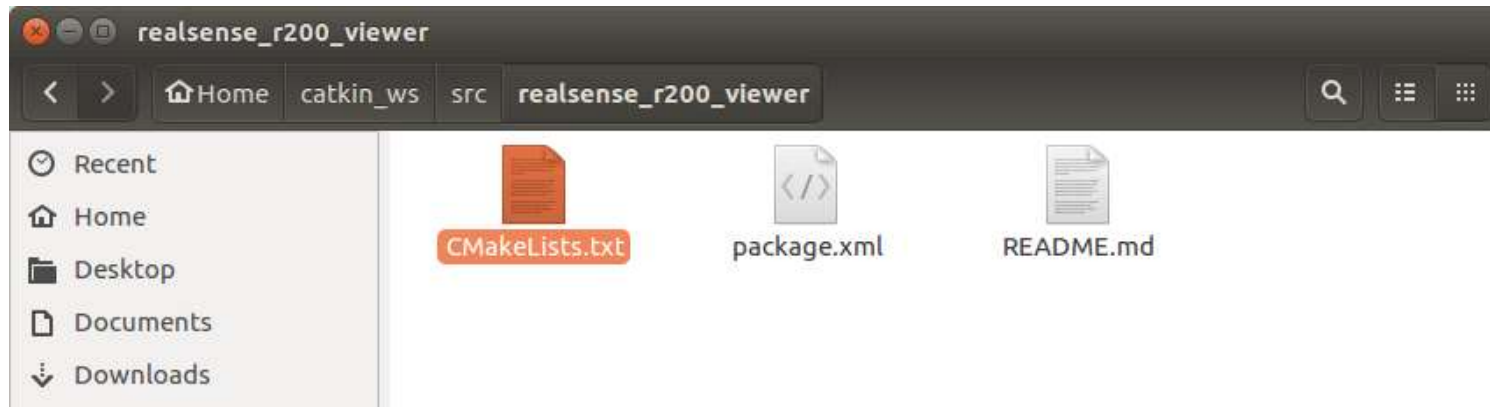
```
bloisi@bloisi-U36SG:~$ rospack find realsense_r200_viewer
/home/bloisi/catkin_ws/src/realsense_r200_viewer
bloisi@bloisi-U36SG:~$
```

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

<http://wiki.ros.org/ROS/Tutorials/Creating%20a%20Package%20by%20Hand>

CMakeList

We need the CMakeLists.txt file so that catkin_make, which uses CMake for its more powerful flexibility when building across multiple platforms, builds the package



<http://wiki.ros.org/ROS/Tutorials/Creating%20a%20Package%20by%20Hand>

CmakeList.txt

```
cmake_minimum_required(VERSION 2.8.3)
project(realsense_r200_viewer)

set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=c++11")

find_package(catkin REQUIRED COMPONENTS
  cv_bridge
  image_transport
  roscpp
  rospy
  sensor_msgs
  std_msgs
)

catkin_package()

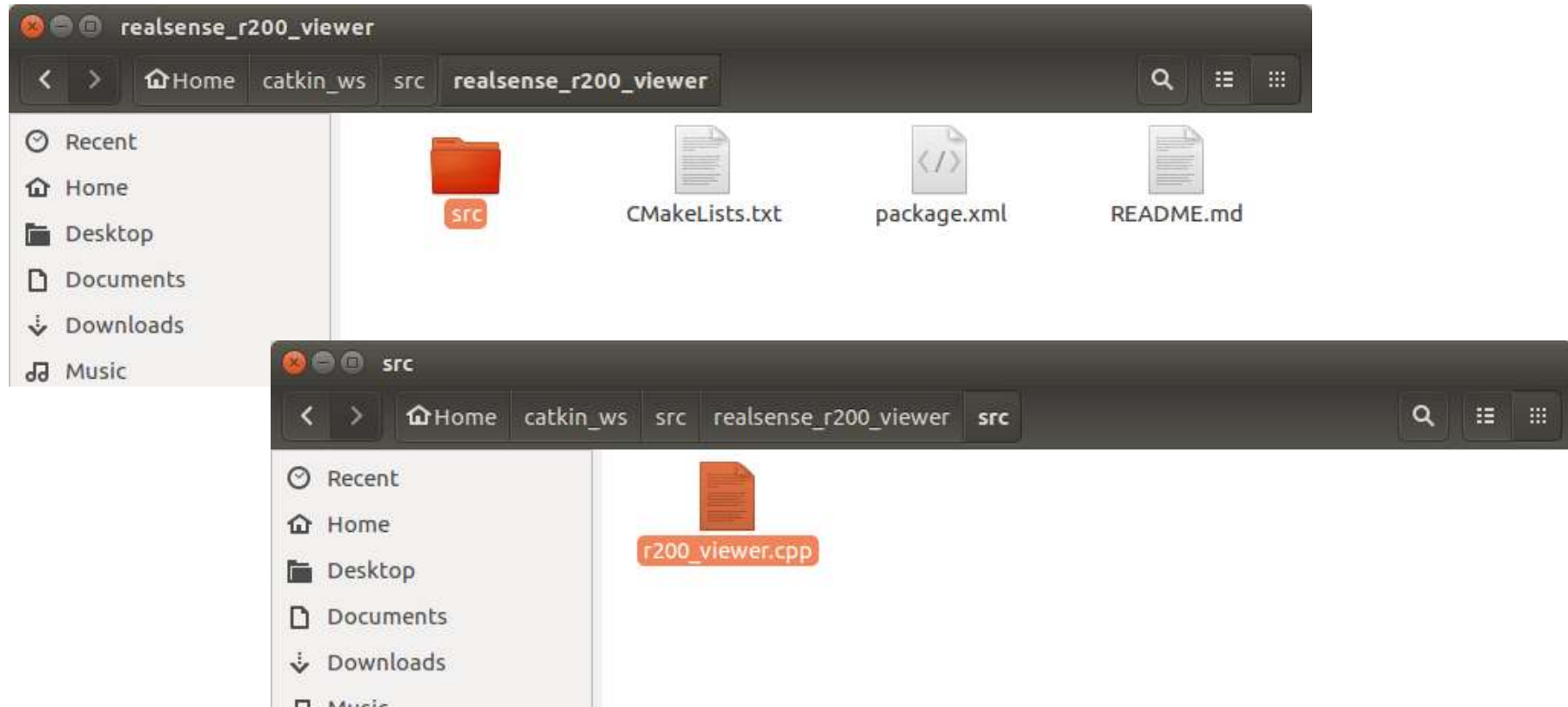
include_directories(
  src/
  ${catkin_INCLUDE_DIRS}
)

set(SRCS src/r200_viewer.cpp)
add_executable(realsense_r200_viewer ${SRCS})
target_link_libraries(realsense_r200_viewer
  ${catkin_LIBRARIES}
)
```

il codice andrà in
src/r200_viewer.cpp



Creazione di src/r200_viewer.cpp



r200_viewer.cpp

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

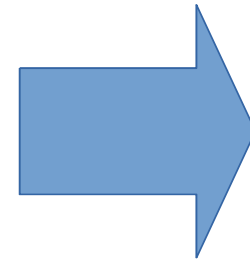
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/imgcodecs.hpp>

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;

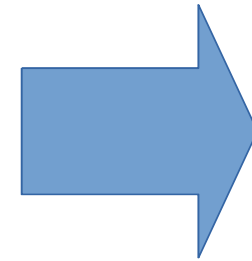
    try {
        cv_ptr = cv_bridge::toCvCopy(msg, msg->encoding);
    } catch (cv_bridge::Exception& e) {
        ROS_ERROR("cv_bridge exception: %s", e.what());
        return;
    }

    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);

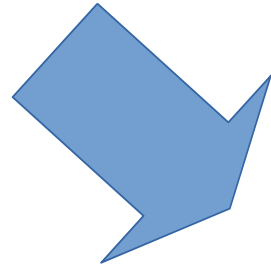
    ros::spin();

    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$
```

roscore + rosrun

Apriamo un terminale e lanciamo `roscore`

```
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]
```

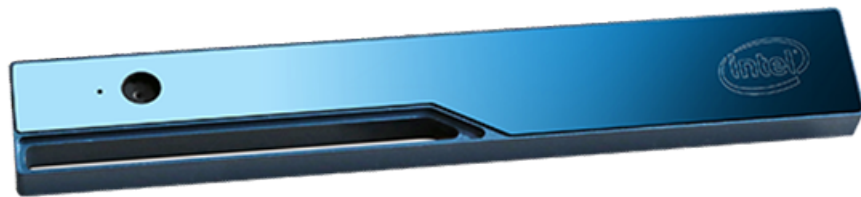
Apriamo un secondo terminale e lanciamo

```
rosrun realsense_r200_viewer
realsense_r200_viewer
```

```
bloisi@bloisi-U36SG:~/catkin_ws
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
```

Cosa accade?

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

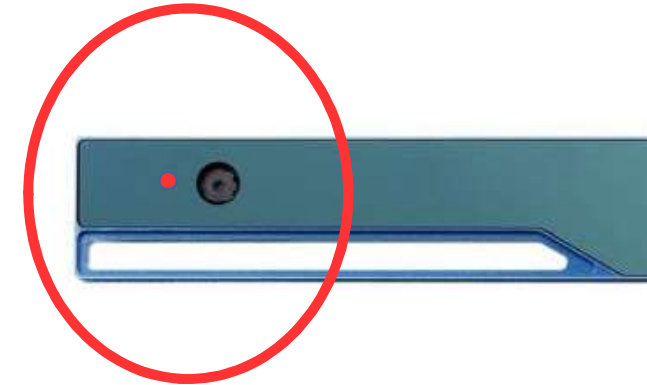


- With the help of a laser projector, the camera does 3D scanning for scene perception and enhanced photography

realsense_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 Serial
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
```



roslaunch (riproviamo)

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

The screenshot shows a ROS environment. A terminal window at the top left displays the command `roslaunch realsense_r200_viewer realsense_r200_viewer` and its output, including subscription topics and "init done". The main window is split into two panels. The left panel shows a standard RGB camera feed of a man with glasses. The right panel shows a depth map of the same scene, where the man's face and glasses are highlighted in red against a dark background. The terminal window at the bottom displays log messages from the camera driver, including:

```
(x=431,y=177) ~ R:179 G:202 B:207
43 <param name="/camera/driver" value="Setting dynamic camera options" type="string"/>
44 <param name="/camera/driver" value="Setting dynamic camera options" type="string"/>
45 </node> [INFO] [1511976924.021751070]: /camera/driver - Setting dynamic camera options
46
47 </launch>
48
```

Aggiungiamo il codice sorgente

git add

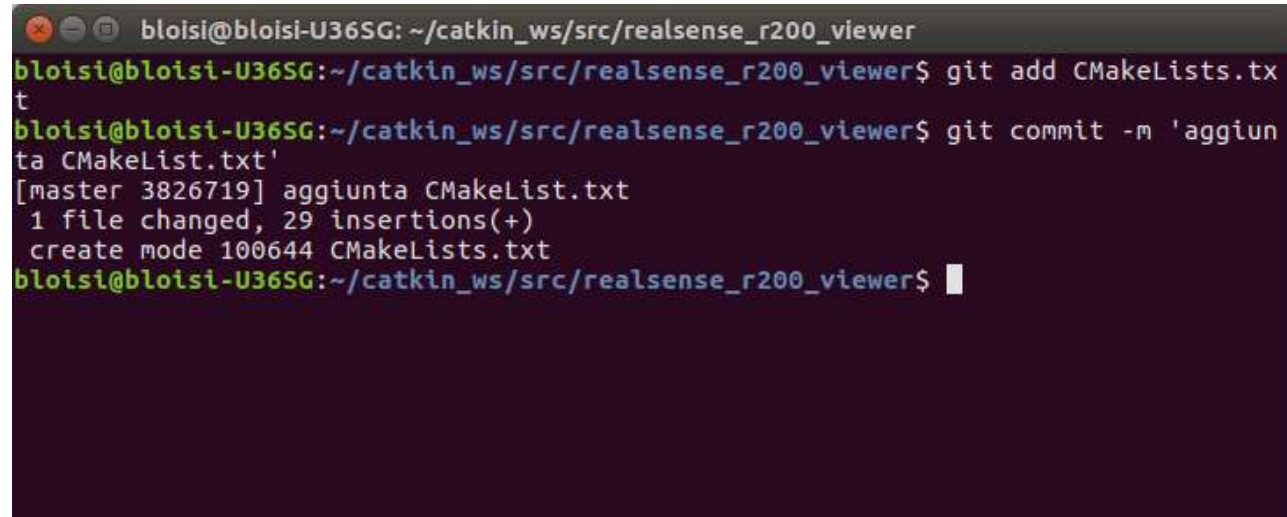
git commit

```
bloisi@bloisi-U36SG: ~/catkin_ws/src/realsense_r200_viewer
bloisi@bloisi-U36SG:~$ cd catkin_ws/src/realsense_r200_viewer/
bloisi@bloisi-U36SG:~/catkin_ws/src/realsense_r200_viewer$ git add src
bloisi@bloisi-U36SG:~/catkin_ws/src/realsense_r200_viewer$ git commit -m 'codice
sorgente'
[master e142f7f] codice sorgente
1 file changed, 73 insertions(+)
create mode 100644 src/r200_viewer.cpp
bloisi@bloisi-U36SG:~/catkin_ws/src/realsense_r200_viewer$
```

Aggiungiamo CmakeLists.txt

```
git add
```

```
git commit
```



```
bloisi@bloisi-U36SG: ~/catkin_ws/src/realSense_r200_viewer
bloisi@bloisi-U36SG:~/catkin_ws/src/realSense_r200_viewer$ git add CMakeLists.txt
bloisi@bloisi-U36SG:~/catkin_ws/src/realSense_r200_viewer$ git commit -m 'aggiunta CMakeList.txt'
[master 3826719] aggiunta CMakeList.txt
1 file changed, 29 insertions(+)
 create mode 100644 CMakeLists.txt
bloisi@bloisi-U36SG:~/catkin_ws/src/realSense_r200_viewer$
```


Aggiorniamo il repo

git push

```
bloisi@bloisi-U36SG: ~/catkin_ws/src/realSense_r200_viewer
bloisi@bloisi-U36SG:~/catkin_ws/src/realSense_r200_viewer$ git push
warning: push.default is unset; its implicit value has changed in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the traditional behavior, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': labrobotica-bloisi
Password for 'https://labrobotica-bloisi@github.com':
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.46 KiB | 0 bytes/s, done.
Total 7 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/labrobotica-bloisi/realSense_r200_viewer.git
   c3aab1e..e142f7f  master -> master
bloisi@bloisi-U36SG:~/catkin_ws/src/realSense_r200_viewer$
```

Aggiungere collaboratori

The screenshot shows the GitHub interface for the repository 'labrobotica-bloisi/realSense_r200_viewer'. The 'Settings' tab is selected and highlighted with a red circle. In the left sidebar, the 'Collaborators' option is also highlighted with a red circle. The main content area shows the 'Collaborators' section with the text: 'This repository doesn't have any collaborators yet. Use the form below to add a collaborator.' Below this, there is a search input field and an 'Add collaborator' button. The footer contains copyright information for GitHub, Inc. and various links like 'Terms', 'Privacy', 'Security', 'Status', 'Help', 'Contact GitHub', 'API', 'Training', 'Shop', 'Blog', and 'About'.

Ricerca del collaboratore

The screenshot shows the GitHub interface for the repository 'labrobotica-bloisi / realsense_r200_viewer'. The 'Settings' tab is active, and the 'Collaborators' section is selected in the left sidebar. The main content area displays the 'Collaborators' settings, including a search input field with the text 'dbloisi' and a dropdown menu showing a suggestion for the user 'dbloisi'. A red oval highlights this search and suggestion area. The page also shows navigation links for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Insights', and 'Settings'. The footer contains copyright information for GitHub, Inc. and various links like 'Terms', 'Privacy', 'Security', 'Status', 'Help', 'Contact GitHub', 'Training', 'Shop', 'Blog', and 'About'.

Invito recapitato al collaboratore

GitHub



@labrobotica-bloisi has invited you to collaborate
on the
labrobotica-bloisi/realSense_r200_viewer
repository

You can [accept or decline](#) this invitation. You can also head over to https://github.com/labrobotica-bloisi/realSense_r200_viewer to check out the repository or visit [@labrobotica-bloisi](#) to learn a bit more about them.

[View invitation](#)

ch Pull requests Issues Marketplace Explore

[realSense_r200_viewer](#)



labrobotica-bloisi invited you to collaborate

[Accept invitation](#)

[Decline](#)

Is this user sending spam or malicious content? You can [block @labrobotica-bloisi](#).

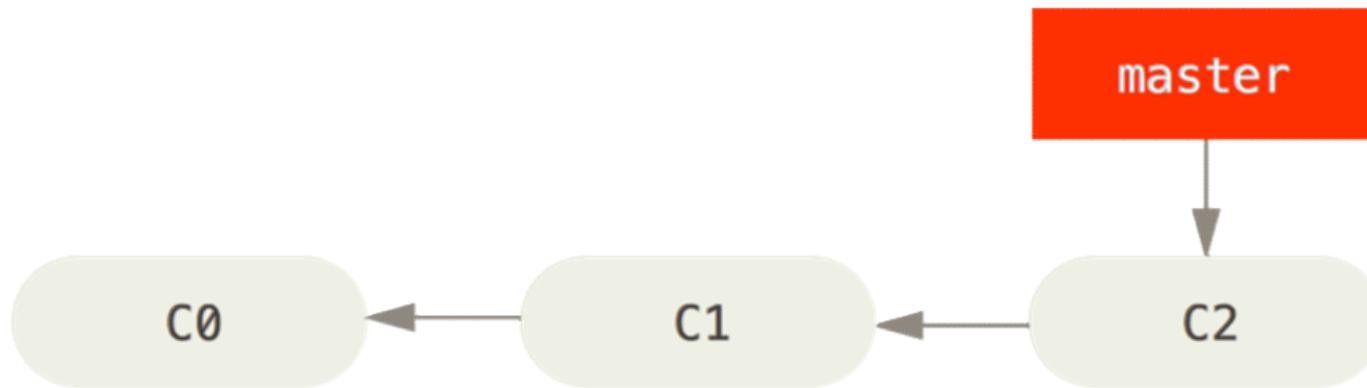
Vista del collaboratore

The screenshot shows the GitHub interface for the repository 'labrobotica-bloisi/realSense_r200_viewer'. At the top, there is a navigation bar with the GitHub logo, a search box, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. A notification banner states: 'You now have push access to the labrobotica-bloisi/realSense_r200_viewer repository.' Below this, the repository name is displayed with icons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). A secondary navigation bar includes 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', and 'Insights'. The repository description is 'ROS node for visualizing data coming from an Intel RealSense R200 device'. A summary bar shows '4 commits', '1 branch', '0 releases', and '1 contributor'. Action buttons include 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The commit history table is as follows:

Commit	Author	Time
dbloisi codice sorgente	dbloisi	Latest commit e142f7f 14 minutes ago
src	codice sorgente	14 minutes ago
README.md	Initial commit	3 hours ago
package.xml	adding manifest file	2 hours ago

Below the table, a 'README.md' file is partially visible.

branching

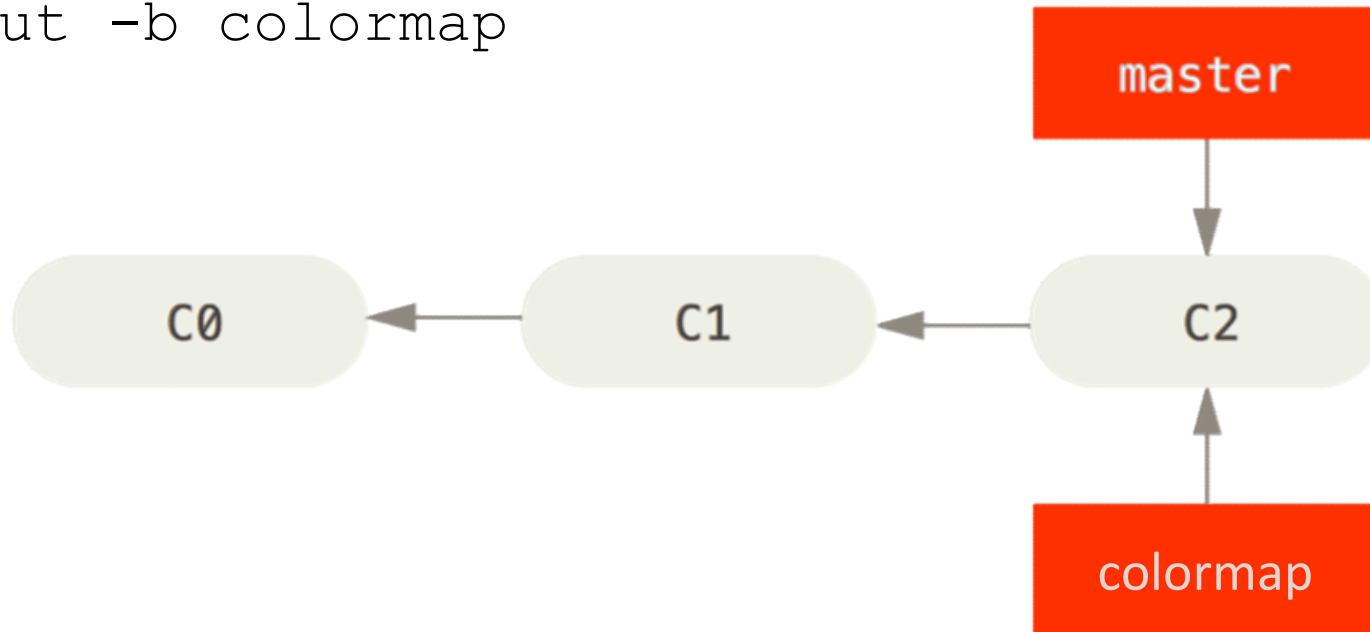


```
git checkout -b colormap
```

<https://git-scm.com/book/it/v2/Git-Branching-Basic-Branching-and-Merging>

branching

```
git checkout -b colormap
```



<https://git-scm.com/book/it/v2/Git-Branching-Basic-Branching-and-Merging>

Modifica a 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();

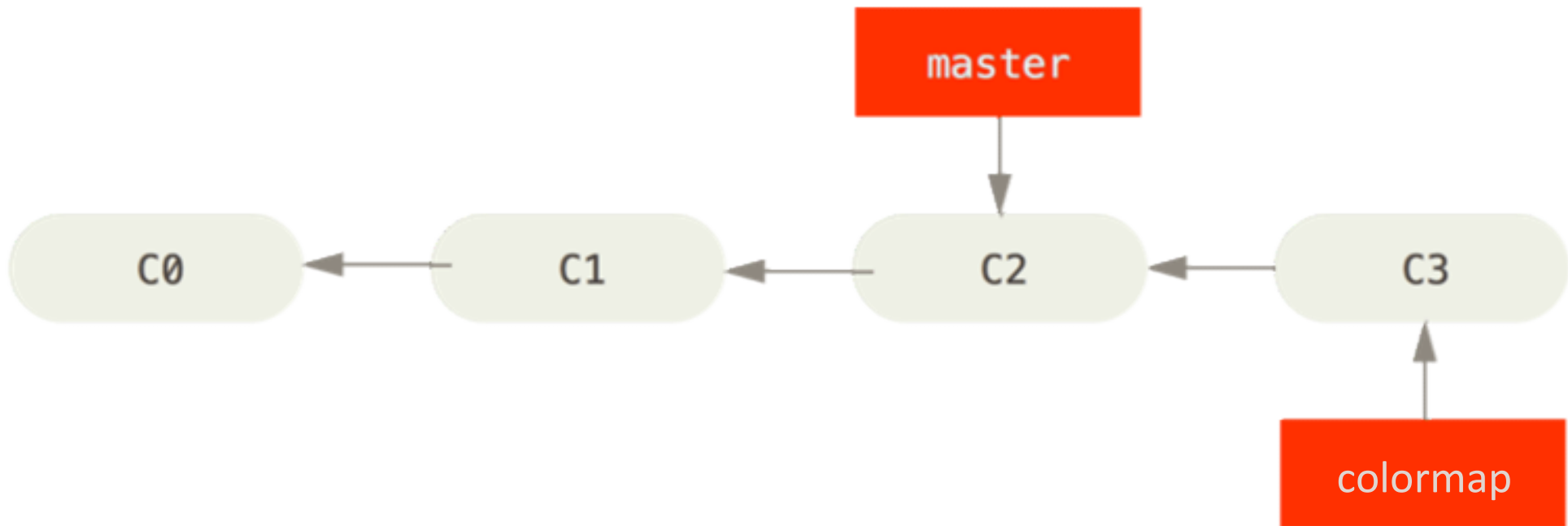
    double dmin, dmax;
    cv::minMaxIdx(depthMat, &dmin, &dmax);
    cv::Mat adjMat;
    cv::convertScaleAbs(depthMat, adjMat, 255 / dmax);

    cv::Mat colorMat;
    cv::applyColorMap(adjMat, colorMat, cv::COLORMAP_JET);
    cv::imshow("Depth", colorMat);

    cv::waitKey(30);
}
```


branching

```
git commit -am 'cambio color map'
```

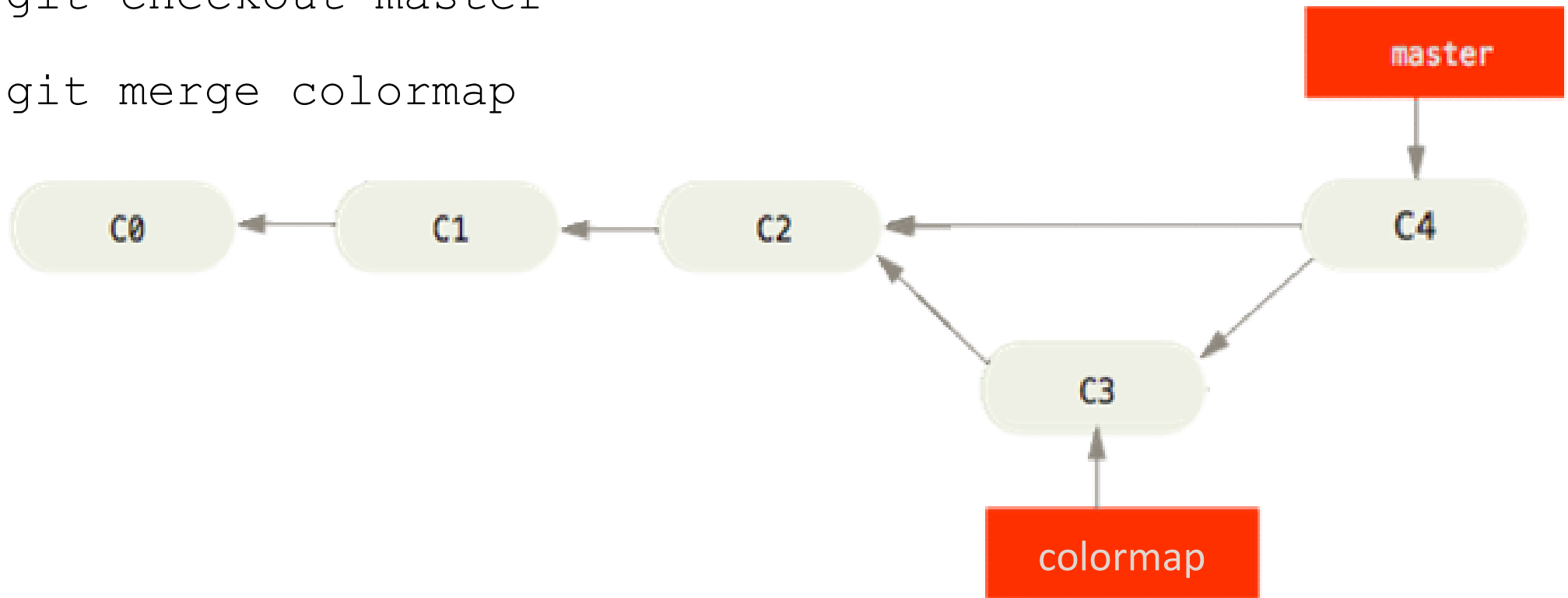


<https://git-scm.com/book/it/v2/Git-Branching-Basic-Branching-and-Merging>

Merging

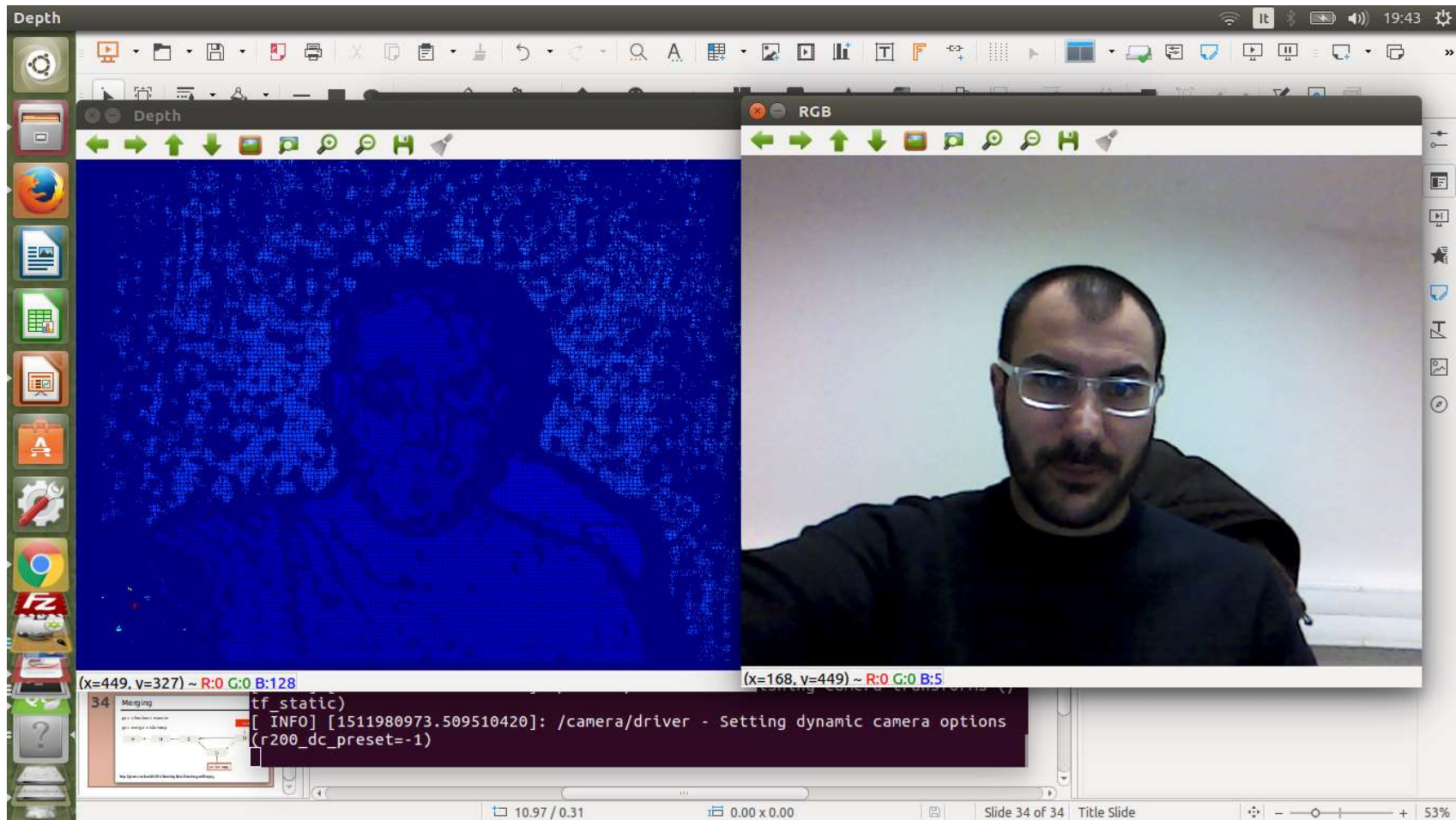
```
git checkout master
```

```
git merge colormap
```



<https://git-scm.com/book/it/v2/Git-Branching-Basic-Branching-and-Merging>

Color map JET



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

Publicare il codice usando un repository Git



UNIVERSITÀ
di **VERONA**

Dipartimento
di **INFORMATICA**

Laurea magistrale in ingegneria e scienze informatiche

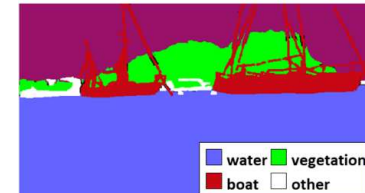
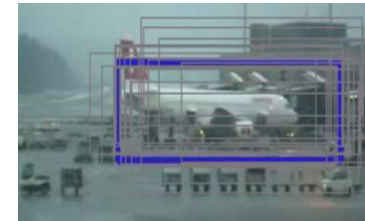
 **ROS** +
 **git**



*Corso di Robotica
Parte di Laboratorio*

Docente:

Domenico Daniele Bloisi



Novembre 2017

