

Camera, Kinect e ROS

Andrey Masiero

5 de fevereiro de 2016

1 Utilizando Webcam e OpenCV

- Segmentação de Cor
- Detecção Facial

2 Exercício

3 Utilizando Kinect

- Controlando a Distância

4 Exercício

5 Apêndice

- Instalação do OpenCV
- Instalação do OpenNI

Primeiro, deve-se instalar o pacote para acessar as informações da webcam:

```
$ sudo apt-get install ros-indigo-uvic-camera
```

Vamos criar um novo pacote dentro do workspace (`~/catkin_ws/src`):

```
$ catkin_create_pkg ros_aula_03 rospy  
  geometry_msgs sensor_msgs std_msgs  
  cv_bridge tf
```

Agora editar o arquivo `CMakeLists.txt` do nosso pacote para incluir o OpenCV:

```
$ cd ~/catkin_ws/src/ros_aula_03  
$ gedit CMakeLists.txt
```

No arquivo CMakeLists.txt procure por `find_package` e insira a seguinte linha:

```
find_package(OpenCV REQUIRED)
```

Depois procure por `include_directories` e insira a seguinte linha:

```
include_directories(  
    ${OpenCV_INCLUDE_DIRS}  
)
```

Assim, o OpenCV está configurado para funcionar em nosso pacote.

Na seção de apêndice 1, tem os passos para instalação do OpenCV

Na sequência, deve-se criar a pasta `script` e iniciar o arquivo de segmentação de cor. Para isso faça:

```
$ mkdir scripts  
$ cd scripts  
$ gedit color_segmentation.py
```

Copie o código `color_segmentation.py`, disponível para download na Aula 03 através do <http://andreymasiero.com/index.php/cursos/ros/>

Feche o editor e de permissão de acesso através do comando:

```
$ chmod +x color_segmentation.py
```

Agora crie um launch para executar o programa:

```
$ cd ~/catkin_ws/src/ros_aula_03  
$ mkdir launch && cd launch  
$ gedit color.launch
```

```
<?xml version="1.0"?>
<launch>
  <node name="camera" pkg="uvc_camera" type="uvc_camera_node">
    <param name="device" value="/dev/video1" />
  </node>
  <node name="color_segmentation" pkg="ros_aula_03" type="color_segmentation.py"
    output="screen" />
</launch>
```

Feche o editor e execute o launch:

```
$ roslaunch ros_aula_03 color.launch
```

Criar o arquivo de Detecção Facial. Para isso faça:

```
$ cd ~/catkin_ws/src/ros_aula_03/scripts  
$ gedit face_dectetion.py
```

Copie o código `face_detection.py`, disponível para download na Aula 03 através do <http://andreymasiero.com/index.php/cursos/ros/>

Feche o editor e de permissão de acesso através do comando:

```
$ chmod +x face_dectetion.py
```

Agora crie um launch para executar o programa:

```
$ cd ~/catkin_ws/src/ros_aula_03/launch  
$ gedit face.launch
```

```
<?xml version="1.0"?>
<launch>
  <node name="camera" pkg="uvc_camera" type="uvc_camera_node">
    <!-- Para webcam do note usar /dev/video0 para webcam externa usar /dev/
         video1 -->
    <param name="device" value="/dev/video1" />
  </node>
  <node name="face_detection" pkg="ros_aula_03" type="face_detection.py" output="
        screen" />
</launch>
```

Feche o editor e execute o launch:

```
$ roslaunch ros_aula_03 face.launch
```

Exercício 1

Faça a nossa tortuguita virar para a direita ou para esquerda de acordo com a posição da pessoa na tela. Para isso, utilize o nó de segmentação de cor.

Agora deve-se instalar os pacotes para acessar as informações do kinect:

```
$ sudo apt-get install ros-indigo-openni-camera ros-indigo-openni-launch
```

É necessário realizar o download do pacote que realiza o rastreamento de uma pessoa com o kinect, execute:

```
$ cd ~/catkin_ws/src
$ git clone https://github.com/ros-drivers/openni_tracker.git
$ cd ~/catkin_ws
$ catkin_make
```

Na seção de apêndice 2, tem os passos para instalação dos drivers para o Kinect.

Na sequência, deve-se criar a pasta `script` e iniciar o arquivo de Controlando a Distância. Para isso faça:

```
$ cd ~/catkin_ws/src/ros_aula_03/scripts
$ gedit distance_control.py
```

Copie o código `distance_control.py`, disponível para download na Aula 03 através do <http://andreymasiero.com/index.php/cursos/ros/>

Feche o editor e de permissão de acesso através do comando:

```
$ chmod +x distance_control.py
```

Agora crie um launch para executar o programa:

```
$ cd ~/catkin_ws/src/ros_aula_03/launch  
$ gedit distance.launch
```

Controlando a Distância

```
<?xml version="1.0"?>
<launch>
  <include file="$(find openni_launch)/launch/openni.launch" />
  <node name="openni_tracker" pkg="openni_tracker" type="openni_tracker" respawn="
    true" output="screen" args="_camera_frame_id:=openni" />
  <node name="distance_control" pkg="ros_aula_03" type="distance_control.py" output
    ="screen"/>
</launch>
```

Feche o editor e execute o launch:

```
$ roslaunch ros_aula_03 distance.launch
```

Faça o controle da tortuguita utilizando o controle de distância com kinect.

Em um terminal, execute os seguintes comandos para garantir a instalação dos pacotes necessários ao OpenCV:

```
$ sudo apt-get install build-essential
$ sudo apt-get install cmake git libgtk2.0-dev
  pkg-config libavcodec-dev libavformat-dev
  libswscale-dev
$ sudo apt-get install python-dev python-numpy
  libtbb2 libtbb-dev libjpeg-dev libpng-dev
  libtiff-dev libjasper-dev libdc1394-22-dev
```

Instalação do OpenCV

Em um terminal, execute os seguintes comandos para a instalação do OpenCV:

```
$ cd ~/Downloads
$ wget https://github.com/Itseez/opencv/
  archive/2.4.11.zip
$ mv 2.4.11.zip opencv.zip
$ unzip opencv.zip
$ mkdir opencv-makefiles
$ cd opencv-makefiles
$ cmake ../opencv-2.4.11
$ make
$ make install
```

Os passos da instalação estão descritos no link

http://docs.opencv.org/2.4/doc/tutorials/introduction/linux_install/linux_install.html

Instalação do Openni

Em um terminal, execute os seguintes comandos para a instalação do Openni e o driver do Kinect:

```
$ cd ~/Downloads
$ wget https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/simple-openni/OpenNI_NITE_Installer-Linux64-0.27.zip
$ unzip OpenNI_NITE_Installer-Linux64-0.27.zip
$ cd OpenNI_NITE_Installer-Linux64-0.27/NITE-Bin-Dev-Linux-x64-v1.5.2.21/
$ sudo ./install.sh
$ cd ../OpenNI-Bin-Dev-Linux-x64-v1.5.4.0/
$ sudo ./install.sh
$ cd ../kinect/Sensor-Bin-Linux-x64-v5.1.2.1/
$ sudo ./install.sh
```