

# 情報計測学基礎 I 課題

T11710M 鳥居拓耶

【課題】 オプティカルフローを用いた物体検出プログラムの開発とインストールマニュアルの制作

## 【インストールマニュアル】

\* 移動物体を検出するプログラムは、T11707M の篠原君の Github にアップロードされているプログラムを参考に作成。

### 1. OpenCV の環境構築とプログラムのビルド (cv\_bridge パッケージの準備)

- 1) パッケージを保存するフォルダの作成

```
mkdir -p my_robot/src
```

- 2) 作成するパッケージ名と依存するパッケージを指定する.

```
cd my_robot/src
```

```
catkin_create_pkg image_converter sensor_msgs cv_bridge roscpp std_msgs  
image_transport
```

※ \_ : 作成するパッケージ名, \_ : 依存するパッケージ

- 3) ソースコード (移動物体を検出するプログラム) を作成する.

```
cd ~my_robot/src/my_opencv/src
```

```
gedit image_optical.cpp
```

下記の URL の Github 内にあがっている image\_optical.cpp ファイルのダウンロード

[https://github.com/t11710m-chukyo/My1stRepository/tree/master/src/my\\_opencv/src](https://github.com/t11710m-chukyo/My1stRepository/tree/master/src/my_opencv/src)

- 4) プログラムをビルドするために、必要なファイルの設定

Github にアップロードされているファイル (labeling.h\_no\_hozon\_basyo) を「~my\_robot/src/my\_opencv/src」にダウンロードし、ファイル (Labeling.h) を「labeling.h\_no\_hozon\_basyo」に記載されているパスにダウンロードする.

- 5) My\_opencv フォルダ内にある CMakeList.txt に以下の 2 行を追加する.

```
add_executable( image_converter src/image_optical.cpp )
```

```
target_link_libraries(image_converter ${catkin_LIBRARIES} )
```

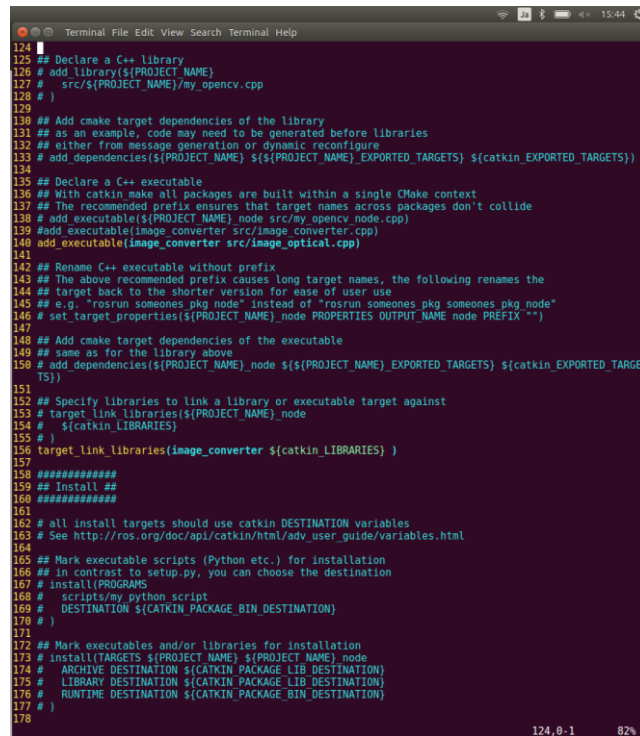
- 6) パッケージ先のフォルダ内の src フォルダで catkin\_init\_workspace を実行する.

```
cd ~my_opencv/src
```

```
catkin_init_workspace
```

- 7) パッケージ先のフォルダ内で `catkin_make` を実行する.

```
cd ~my_opencv
catkin_make
```



```
124
125 ## Declare a C++ library
126 # add_library(${PROJECT_NAME})
127 #   src/${PROJECT_NAME}/my_opencv.cpp
128 # )
129
130 ## Add cmake target dependencies of the library
131 ## as an example, code may need to be generated before libraries
132 ## either from message generation or dynamic reconfigure
133 # add_dependencies(${PROJECT_NAME} ${${PROJECT_NAME}_EXPORTED_TARGETS} ${catkin_EXPORTED_TARGETS})
134
135 ## Declare a C++ executable
136 ## With catkin make all packages are built within a single CMake context
137 ## The recommended prefix ensures that target names across packages don't collide
138 # add_executable(${PROJECT_NAME}_node src/my_opencv_node.cpp)
139 # add_executable(image_converter src/image_converter.cpp)
140 add_executable(image_converter src/image_optical.cpp)
141
142 ## Rename C++ executable without prefix
143 ## The above recommended prefix causes long target names, the following renames the
144 ## target back to the shorter version for ease of user use
145 ## e.g. "roslaunch someones_pkg node" instead of "roslaunch someones_pkg someones_pkg_node"
146 # set_target_properties(${PROJECT_NAME}_node PROPERTIES OUTPUT_NAME node PREFIX "")
147
148 ## Add cmake target dependencies of the executable
149 ## same as for the library above
150 # add_dependencies(${PROJECT_NAME}_node ${${PROJECT_NAME}_EXPORTED_TARGETS} ${catkin_EXPORTED_TARGETS})
151
152 ## Specify libraries to link a library or executable target against
153 # target_link_libraries(${PROJECT_NAME}_node
154 #   ${catkin_LIBRARIES}
155 # )
156 target_link_libraries(image_converter ${catkin_LIBRARIES} )
157
158 #####
159 ## Install ##
160 #####
161
162 # all install targets should use catkin DESTINATION variables
163 # See http://ros.org/doc/api/catkin/html/adv_user_guide/variables.html
164
165 ## Mark executable scripts (Python etc.) for installation
166 ## in contrast to setup.py, you can choose the destination
167 # install(PROGRAMS
168 #   scripts/my_python_script
169 #   DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
170 # )
171
172 ## Mark executables and/or libraries for installation
173 # install(TARGETS ${PROJECT_NAME} ${PROJECT_NAME}_node
174 #   ARCHIVE DESTINATION ${CATKIN_PACKAGE_LIB_DESTINATION}
175 #   LIBRARY DESTINATION ${CATKIN_PACKAGE_LIB_DESTINATION}
176 #   RUNTIME DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
177 # )
178
```

図 1 修正後の CMakeList.txt

## 2. USB カメラの使用準備

- 1) `usb_cam` パッケージのダウンロード

```
cd ~my_opencv/src
git clone https://github.com/bosch-ros-pkg/usb\_cam.git
```

- 2) `usb_cam` パッケージの `make`

```
cd ~my_opencv
catkin_make
```

## 3. プログラムの実行方法

- 1) USB カメラを PC に接続する.
- 2) 3 つのターミナルを用意し, `my_opencv` まで移動した後, `setup.bash` を実行する.

```
cd ~my_opencv
source devel/setup.bash
```

- 3) 各ターミナルで以下のコマンドの 1 つを実行する. (すべてのコマンドを実行する必要有り)

Roscore

//

roslaunch usb\_cam usb\_cam\_node

roslaunch my\_opencv image\_converter

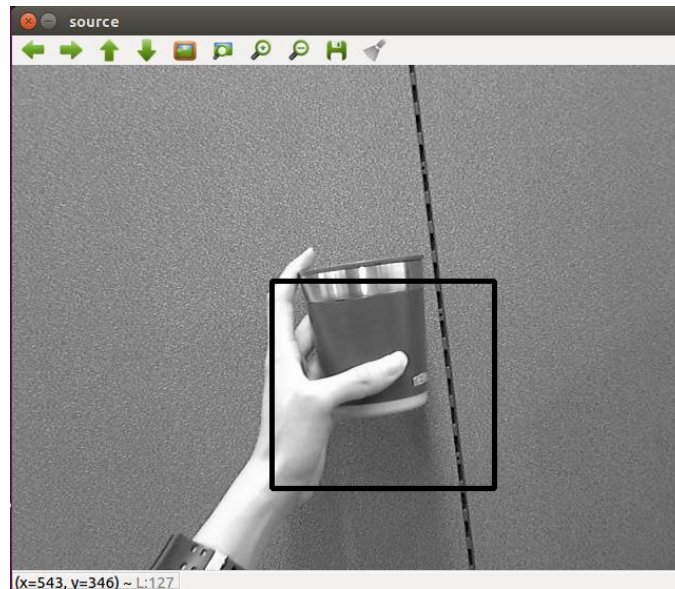


図 2 移動物体の検出結果の例

ウインドウ上の移動物体に対してバウンディングボックスが囲われる.