

RoboVision

This page is **stale** and needs to be reviewed. It may be deleted or radically changed in the near future.

Due to the difficulty in setting up, testing, and tweaking computer vision algorithmic pipelines using OpenCV alone, the RoboVision GUI was made to make this process extremely fast and easy. Before moving on, you should read the vision system wiki to get a better understanding of the underlying data structures and use cases of this tool.

Introduction

The RoboVision GUI is a custom C++ Qt application that utilizes the vision system's FilterTree and Filter objects as it's underlying base. It's main use case is to be able to load an arbitrary video and filtering tree, see the entire pipeline at once, tweak values as required, and save to be able to run in the vision system with it's new parameters. In addition to these, it also provides useful functions such as Add, Edit, and Delete nodes, and useful displays such as a video scrubber, and runtime and tree information.

Building RoboVision

An automated build script is present in the scripts folder which builds all repo GUI tools.

For a custom build you need a Qt5 install as well as QtCreator on a Linux machine. To load the project, open the `robovision.pro` file in `gui/build/robovision`. Once that is loaded you will need to set the build and run paths. For build path, it should be `bin/robovision/debug` and `bin/robovision/release` for debug and release respectfully. For the run path it should be the top level `robosub` install directory as all programs should run at the top level to be able to gather resources properly. Once those are configured run QMake from the build menu then **Build** and **Run**. The application should startup after a few seconds of building.

Using RoboVision

Loading Video

Using RoboVision should be somewhat intuitive as it's a very simple tool that mirrors the vision system. To load a video click Load at the top right corner next to the video scrubber. It should only list compatible video codecs in the file explorer. (NOTE: Due to the nature of OpenCV's VideoCapture object, some codecs will not run or may even crash the tool. This is due to lack of support for some codecs in OpenCV's source code and can't be fixed unless we change libraries for video objects.) Once loaded you should see the first frame of the video in the center of your screen.

Loading Trees

To Load a filtering tree you can select **Open** in the file menu or **Ctrl-O**. It should only list `.xml` files in the file explorer. Some sample trees should be in `settings/vision/trees/`. If it properly loads you should see windows appear for each filter and the filters in the tree-view panel on the right side.

Playing with Video

The application provides useful video scrubbing options in the top right. (NOTE: Once again, due to the lack of some video codec support, some video encodings may not work correctly. If a codec is not working, try to convert to an official format using VLC.) To start or stop a video, press play or stop in the center of the video player. The default framerate should be 20 fps. To step forward or back one frame you can use the greater than / less than sign buttons. The shift left / shift right do the same thing but for 10 frames. You can also drag the video scrubber to get to a part of the video that you want to see. Once a video finishes, it should resume from the beginning.

Playing with Trees

To edit tree values all you have to do is select the filter you want to edit and it's parameters will show up on the bottom panel. The frame that represents that filter's output should also be highlighted in red in the pipeline viewer. You can add, edit, and delete filters that you have selected as well using the appropriate buttons. When changing or adding filters you have the option of making it a child or sibling just like you would in the vision system. You also have the option of giving it a unique name.

Saving

To save your tree all you have to do is click **Save** or **Save As** in the file menu or **Ctrl-S**. Your tree should be updated from the load or last saved location.

FAQ

- The RoboVision app won't build properly.
 - Make sure that you have all dependencies installed correctly and configured the project as explained. If it still gives problems, try cleaning the project. If that doesn't work, try deleting the build folder and rerunning `qmake`. If that doesn't work, tough shit.
- My video won't load or play properly.
 - Read the video section as it explains why this issue exists.
- The sexy orange stylesheet isn't being loaded. The app looks horrible.
 - Make sure the run directory is the top level of the robosub repo so it can access it's resources properly.
- The project configuration prompt detects no Qt Kit.
 - If it doesn't auto-detect your install, click **Options** and then navigate to the `qmake` binary in the `bin` folder of your install.
- The cascading else if's for the `filtertree` is the ugliest code I have ever seen, what is wrong with

you?

- The amount of work it would take to make the project fully dynamic for the serialization and deserialization of new, user created filters is not something I have the time or work ethic to finish right now. Make your own damn vision tool.

From:

<http://robosub-vm.eecs.wsu.edu/wiki/> - **Palouse RoboSub Technical Documentation**

Permanent link:

<http://robosub-vm.eecs.wsu.edu/wiki/legacy/2016/cs/robovision/start?rev=1475180390> 

Last update: **2016/09/29 13:19**