Royale 5.12.0.3089

Introduction

Version 5.12.0.3089

5.12.0.3089
Created: 22.08.2024

Contents

1 Introduction

The **Royale** software package provides a light-weight camera framework for time-of-flight (ToF) cameras. While being tailored to PMD cameras, the framework enables partners and customers to evaluate and/or integrate 3D TOF technology on/in their target platform. This reduces time to first demo and time to market.

Royale contains all the logic which is required to operate a ToF based camera. The user doesn't need to care about setting registers, but can conveniently control the camera via a high-level interface. The Royale framework is completely designed in C++ using the C++11 standard.

1.1 Operating Systems

Royale supports the following operating systems:

- Windows 10
- Linux (tested on Ubuntu 20.04)
- Android (tested on Android 8 (ARMv8a with NEON))
- Linux ARM (32Bit version tested on Raspbian GNU/Linux 10 (Buster) Raspberry Pi reference 2020-08-20 (hard float ABI) 64Bit version tested on a Raspberry 4 with Ubuntu 20.04 ARM 64 (hard float ABI))

1.2 Hardware Requirements

Royale is tested on the following hardware configurations:

- PC, AMD Ryzen 7 PRO 5850U (64 bit)
- Samsung Galaxy S9
- Raspberry Pi 4
- On x86 platforms : AVX2 support required
- On ARMv7A platforms : NEON >= v4 required

2 Getting Started

For a detailed guide on how to get started with Royale and your camera please have a look at the corresponding Getting Started Guide that can be found in the top folder of the package you received.

3 SDK and Examples

Besides the royaleviewer application, the package also provides a Royale SDK which can be used to develop applications using the PMD camera.

There are multiple samples included in the delivery package which should give you a brief overview about the exposed API. You can find an overview in samples/README_samples.md. The *doc* directory offers

a detailed description of the Royale API which is a good starting point as well. You can also find the API documentation by opening the API_Documentation.html in the topmost folder of your platform package.

The easiest way to use the SDK in your project is with CMake. If you use CMake, add directory "<Royle_Installation_Prefix>/lib/cmake" to your "CMAKE_PREFIX_PATH" and and add "target_link_library(<target-using-royale> PRIVATE royale::royale)" to link libroyale against your target.

3.1 Debugging in Microsoft Visual Studio

To help debugging royale::Vector, royale::Pair and royale::String we provide a Natvis file for Visual Studio. Please take a look at the natvis.md file in the doc/natvis folder of your installation.

4 Matlab

In the delivery package for Windows you will find a Matlab wrapper for the Royale library. After the installation it can be found in the matlab subfolder of your installation directory. To use the wrapper you have to include this folder into your Matlab search paths. We also included some examples to show the usage of the wrapper. They can also be found in the matlab folder of your installation.

5 Python

In the package you will also find a wrapper to use Royale with Python. Unfortunately this wrapper will only work with specific Python versions. Please have a look at the README.md file in the Python folder to find out which versions are currently supported. In case you want to use a different Python version you can still use the SWIG interface file from the SWIG folder to compile your own wrapper.

6 Reference

FAQ: https://3d.pmdtec.com/en/support/faq/

7 License

See ThirdPartySoftware.txt and royale_license.txt. The source code of the open source software used in the Royale binary installation is available at https://oss.pmdtec.com/.