Beam Profiling Application - Manual

# Introduction

## About This Manual

The porpuse of this document is to describe a beam profiling application. This application helps to monitor and characterize laser beams.

The manual guides the user through the installation process and describes the features of the program. It also sketches how to make it compatible with other camera types. The software has been tested with a Ximea xiQ camera and several VRmagic USB cameras. There also exists an interface to uEye cameras, which has been tested with an old version of the program.

## Copyright and License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 3 of the License.

This program is distributed in the hope that it will be useful, but **WITHOUT ANY WARRANTY**; without even the implied warranty of **MERCHANTABILITY** or **FITNESS FOR A PARTICULAR PURPOSE**. See the GNU General Public License for more details.

Please see the README.md file for a copy of the GNU General Public License, or otherwise find it on <http://www.gnu.org/licenses/>.

## Contributors

* **Cristian Groß** [christian.gross@mpq.mpg.de](mailto:christian.gross@mpq.mpg.de)
* **Timon Hilker** [timon.hilker@mpq.mpg.de](mailto:timon.hilker@mpq.mpg.de)
* **Michael Höse** [michael.hoese@physik.uni-muenchen.de](mailto:michael.hoese@physik.uni-muenchen.de)
* **Konrad Viebahn** [kv291@cam.ac.uk](mailto:kv291@cam.ac.uk)

For questions on the source code refer to **Michael Höse.** Questions on the uEye interface should be directed to **Konrad Viebahn.**

# Installation

## Preparations

### Python

Make sure that a current version of Python 2.7 is installed on your computer (64-bit recommended). If not, install a Python distribution that contains the relevant scientific packages (Numpy, Scipy, etc.). For testing Anaconda was used. To install Anaconda go to <https://www.continuum.io/downloads>. Download and install the suitable version (64-bit recommended).

### Pyqtgraph

There are two options to get “pyqtgraph” running on your computer

* If “pyqtgraph” is not yet installed on your computer, go to <http://pyqtgraph.org/> and follow the installation instructions
* In case you do not want to install “pyqtgraph” or if any problems occur, you can directly download the source code (as a Zip-folder) or clone into the “pyqtgraph” repository here <https://github.com/pyqtgraph/pyqtgraph>. The following steps will be explained later.

## Necessary Camera Software

### VRmagic USB Cameras

To run the beam profiling software with VRmagic USB cameras, go to <https://www.vrmagic.com/de/imaging/downloads/> and download and install the “VRmUsbCam DevKit”. If your Python installation is 32-bit use the 32-bit version, otherwise download the 64-bit version.

### Ximea xiQ Cameras

When using Ximea xiQ cameras, go to <https://www.ximea.com/support/wiki/apis/APIs> to download and install the “XIMEA API Software Package” that suits your system.

## Retrieving Files from GitHub

The most convenient way to get the necessary files to run the beam profiling application is to clone the Git repository from GitHub. Therefore Git has to be installed on your computer. To clone the repository, open the Git Bash and move to the directory, where you want to clone the repository to. Run the command **git clone https:// github.com/mhoese/beam-cam** in the Git Bash. Then change to the “mpq” branch by calling **git checkout mpq**. Now you should see in the directory you have chosen a new folder “beam-cam”. This folder contains the program files. Instead of cloning, you can also just download the files from <https://github.com/mhoese/beam-cam>.

To run the program with VRmagic USB cameras, you have to copy the “vrmusbcam2.dll” file into the “beam-cam” folder. You find this file in the VRmagic directory (by default in CommonFiles).

When you want to use Ximea xiQ cameras, copy the “xiapi64.dll” to the “beam-cam” folder. This file can be found in the XIMEA directory.

By default, the program loads both of these files, so make sure that both of them are in the “beam-cam” folder. If you only want to use one camera, you will have to specify this in the “BeamProfilingApplication.py” file. For details see the section on how to configurate different camera types.

In case you do not want to install “pyqtgraph”, you have to copy the “pyqtgraph” –folder from the repository of “pyqtgraph” (see section 2.1.2.) into the “beam-cam” folder.

To start the program, run “BeamProfilingApplication.py” with Python.

# Preparations

This chapter specifies how to adapt the program to different camera types. It is also possible to only install the software of one of the two cameras described above. If doing so, some default settings in the source code have to be changed.

## Modifications for Using a Single Camera Type

If only a single camera type (VRmagic **or** Ximea) is used, there is no need to install the software for the other camera type. To run the program, some changes in the source code have to be made. In this part of the source code, comment out the options that do not meet your needs. You find those at the beginning of “BeamProfilingApplication.py”.

'''!!COMMENT OUT THE OPTIONS THAT YOU DO NOT WANT!!'''

'''Using both camera types (VRmagic and Ximea)'''

cameratypes = ['VRmagic USB','Ximea xiQ']

cameraapinames = ['VRmagicUsbCamAPI','XimeaxiQCamAPI']

# '''Using VRmagic cameras only'''

# cameratypes = ['VRmagic USB']

# cameraapinames = ['VRmagicUsbCamAPI']

# '''Using Ximea xiQ cameras only'''

# cameratypes = ['Ximea xiQ']

# cameraapinames = ['XimeaxiQCamAPI']

# '''Only use demo'''

# cameratypes = []

# cameraapinames = []

'''-----------------------------------------------'''

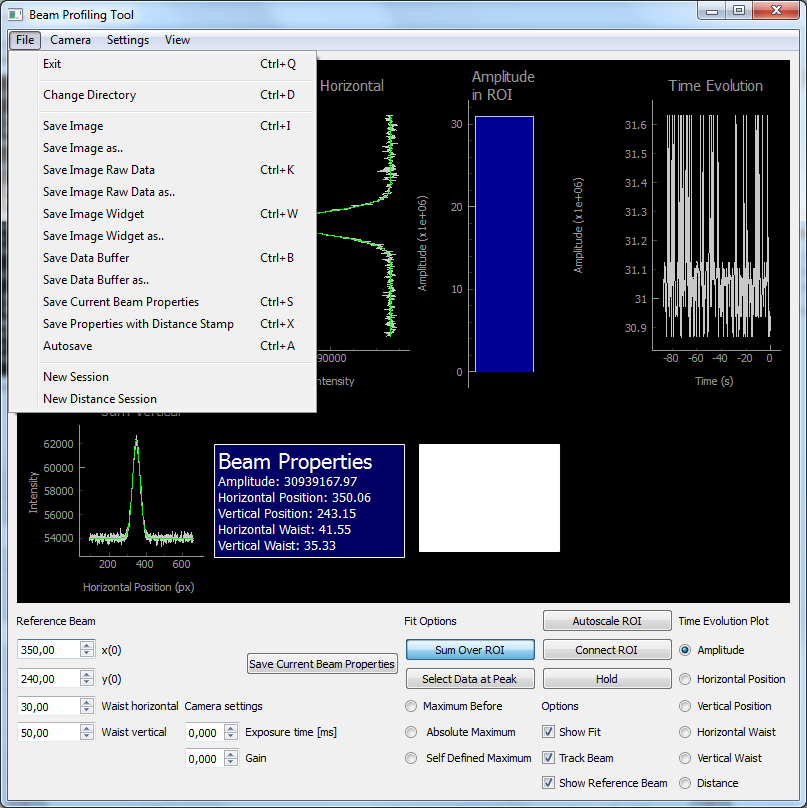
## Modifications for Using Another Camera Type

To use a camera type that is not compatible with any of the available interfaces, a new subclass that extends the methods used in the superclass (contained in “CameraAPI.py”) has to be written. This new interface has to overwrite all the methods specified in the superclass. The datatypes of arguments and the structure of returned values have to match the specifications exactely to ensure a working program. For examples see “XimeaxiQCamAPI.py” or “VRmagicUsbCamAPI.py”. Finally you have to define a new name for the GUI in the list “cameratypes” and the name of the corresponding interface in the list “cameraapinames”. Examples can be seen in section 3.1. These lists can be found at the beginning of the file “BeamProfilingApplication.py”.

# Functions of the Program

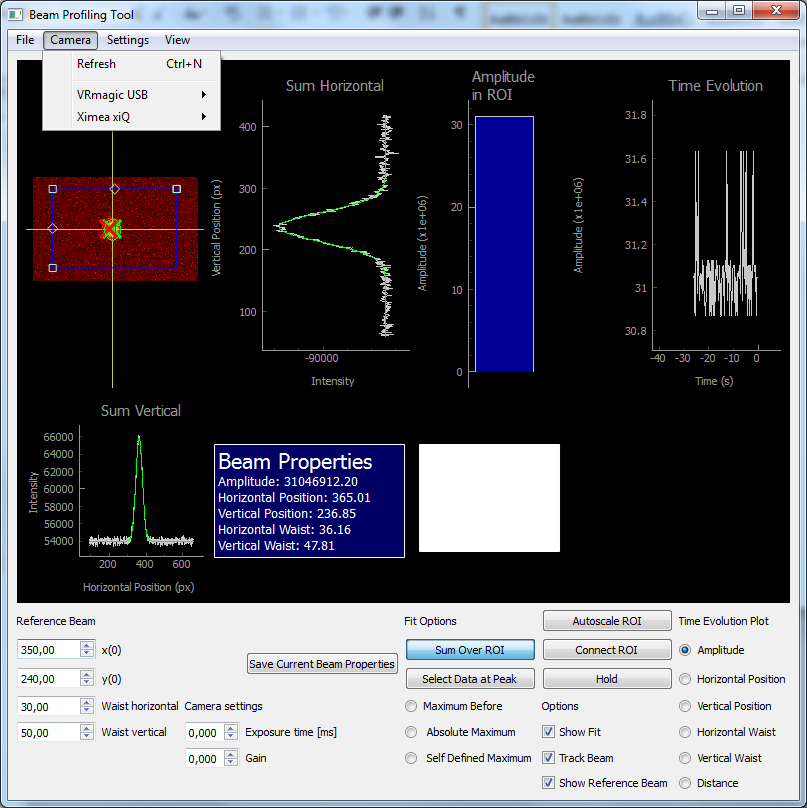
This section gives an overview of the functions of the program. All the options are shortly described.

## Menubar – File



* **Exit:** Quit the application
* **Change Directory:** Change the directory where the data or images are save
* **Save Image:** Save the image as .png file
* **Save Image as:** Save the image as .png file with self-specified name
* **Save Image Raw Data:** Save the image raw data as .npy file
* **Save Image Raw Data as:** Save the image raw data as .npy file with self-specified name
* **Save Image Widget:** Save a part of the displayed interface as .png file
* **Save Image Widget as:** Save a part of the displayed interface as .png file Save the current data buffer as .csv file
* **Save Data Buffer:** Save the current data buffer as .csv file
* **Save Data Buffer as:** Save the current data buffer as .csv file Save the current data buffer as .csv file
* **Save Current Beam Properties:** Creates a .csv file in case it not exists yet and ads a line with the current beam properties (also accessible from the “Save Current Beam Properties” button in the GUI)
* **Save Properties with Distance Stamp:** Same as “Save Current Beam Properties”, but a distance stamp has to be entered manually (possible applications: measure Gaussian beam).
* **Autosave:** If checked: Creates a .csv file and saves the current beam properties after a time interval that has to be specified when starting the function.
* **New Session:** Opens a new .csv file when “Save Current Beam Properties” is used afterwards.
* **New Distance Session:** Opens a new .csv file when “Save Properties with Distance Stamp” is used afterwards.

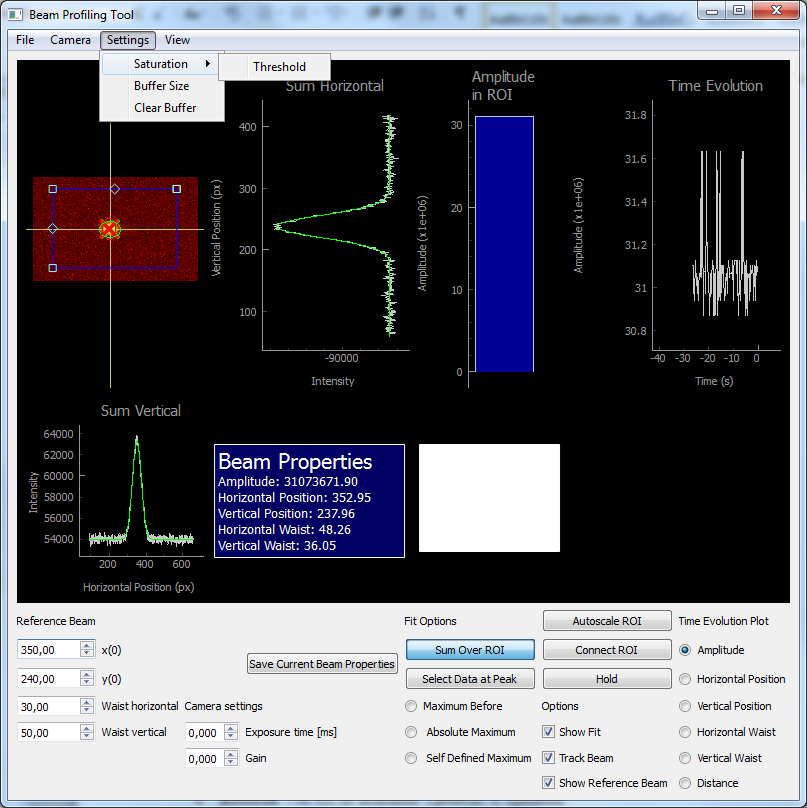
## Menubar – Camera



* **Refresh:** The list of available cameras is updated

Below, the compatible camera types are shown. In the according sub-menus the available cameras are displayed. By clicking on them, the current operating camera is changed.

## Menubar – Settings



* **Saturation – Threshold:** The minimal number of pixel that has to be saturated for the saturation warning to show up can be changed here
* **Buffer Size:** The size of the data buffer can be adjusted here
* **Clear Buffer:** Clear the data buffer

## Menubar – View

