

# Contents

<b>FIMrecorder</b>	<b>1</b>
<b>1 Installation</b>	<b>1</b>
<b>2 Supported Devices</b>	<b>2</b>
<b>3 Overview</b>	<b>2</b>
<b>4 Configuration</b>	<b>3</b>
4.1 settings.json . . . . .	3
4.2 .pfs Files . . . . .	4
<b>5 Basic Workflow</b>	<b>4</b>
5.1 Pre-Recording . . . . .	4
5.1.1 Checking Setup . . . . .	4
5.1.2 Measurement Annotations . . . . .	5
5.1.3 Applying Camera parameters . . . . .	5
5.1.4 Real-Time Signal Modifications . . . . .	6
5.2 Recording . . . . .	6
5.3 Post-Recording . . . . .	6
5.3.1 Locating Recorded Data . . . . .	7
5.3.2 Extracting Frames from Video Files . . . . .	7
<b>6 Troubleshooting</b>	<b>7</b>
6.1 Known Limitations . . . . .	7
6.2 Logging . . . . .	7
6.2.1 loggingconf.json . . . . .	8

## FIMrecorder

*FIMrecorder*<sup>1</sup> is a special-purpose video-recording application for FIM<sup>2</sup> experiments. This software is at the level of a prototype. It works reliably but has only been tested in a small scope (see Supported Devices). The features of this application have been designed to work well in conjunction with FIMTrack<sup>3</sup> <sup>4</sup>.

## 1 Installation

See README.md<sup>5</sup>.

---

<sup>1</sup><https://github.com/fncnt/fimrecorder>

<sup>2</sup><https://www.uni-muenster.de/PRIA/en/FIM/index.html>

<sup>3</sup><https://www.uni-muenster.de/PRIA/en/FIM/download.shtml>

<sup>4</sup><https://github.com/i-git/FIMTrack>

<sup>5</sup><https://github.com/fncnt/fimrecorder/blob/master/README.md#prerequisites>

## 2 Supported Devices

Currently, only Basler USB3 vision cameras recording in Mono8 format are supported and only the model acA1920-40um actually has been tested.

## 3 Overview

Figure 1 provides an overview concerning the UI of *FIMrecorder*.

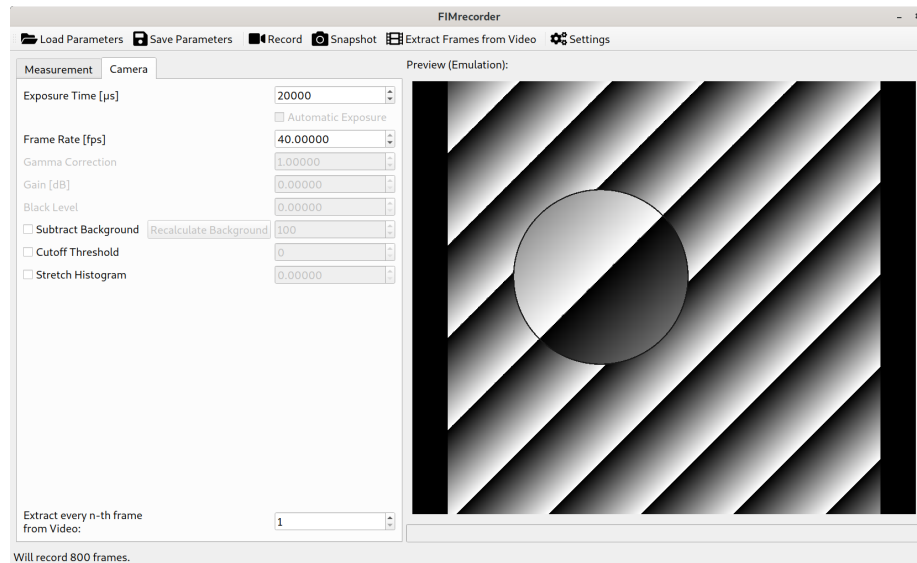


Figure 1: The main UI components of *FIMrecorder*

- (1) Loads parameters ((7) & (8)) from a previously saved `.json` file.
- (2) Saves parameters ((7) & (8)) to a new `.json` file.
- (3) Starts recording for the in (7) specified duration.
- (4) Saves a snapshot to **Snapshot Directory**
- (5) Extracts frames as single images from a specified video file (Required for FIMTrack).
- (6) Starts your favourite text editor to edit the settings file.
- (7) This tab houses information relevant to your experiment such as measurement duration, species or genotype.
- (8) Parameters in this tab modify the image signal your camera is acquiring.
- (9) model name of the camera you're using.
- (10) Live preview of the acquired image signal.
- (11) Zooms by scrolling with your mouse or trackpad.
- (12) When recording or extracting, the progress is displayed here.
- (13) Some relevant status messages appear in this area.

## 4 Configuration

To adjust settings not visible in the UI, click the button labelled *Settings* (6). This will launch your favourite text editor allowing you to edit the main configuration.

### 4.1 settings.json

`settings.json` is the primary configuration file and can be edited using any text editor. It contains data ("**Parameters**") relevant to your measurement (see Measurement Annotations for a detailed description). More importantly, it contains a "**Settings**" section controlling the behaviour of the application. The following options can be modified:

**Background Frames to average:** Number of Frames that should be used to construct an averaged static background image for background subtraction. This setting *is* available from the UI.

default: 100

**Configuration Directory:** Path of the directory where additional configurations files should be stored.

`settings.json` is **not** stored here.

default: "config"

**Default Camera Parameters:** This is the default `.pfs` file. See [#pfsfiles] for further information

default: "FIM\_NodeMap.pfs"

**Extract every n-th Frame:** This value determines how many frames are skipped when extracting from video files.

Usually, a value half the framerate of the recorded footage is a good choice, therefore extracting a frame every 0.5 seconds.

This setting *is* available from the UI.

default: 1

**Logging Configuration:** The location of your logging configuration. There shouldn't be any need to modify this. See [#loggingconf] for more details.

default: "loggingconf.json"

**Recording Directory:** Path of the directory where recorded video and auxiliary files should be stored.

default: "FIMrecordings"

**Single Image Format:** The format snapshots and extracted frames are saved in.

FIMTrack officially supports all three of these, however, ".png" may be problematic for FIMTrack in some cases.

supported: ".tif", ".tiff", ".png"

default: ".tif"

**Snapshot Directory:** The location where snapshots are being saved automatically.

default: "snapshots"

**Video Codec:** The codec in which recorded video files should be saved.

In principle, every `fourCC`<sup>6</sup> installed on your system *and* supported by `ffmpeg` and `opencv` should work.

So far, only "XVID" has been tested and determined to be a safe choice.

default: "XVID"

**Video Container Format:** The container format of your recorded video footage.

Only the default value has been tested. `opencv` has some limitations<sup>7</sup>.

default: ".avi"

**Note:** The default values for directories specify subdirectories of the application relative to itself. It is possible to specify absolute paths, though (e.g. "C:/Users/FIM User/Desktop/FIMrecordings"). Additionally, relative paths are not restricted to subdirectories; "../FIMrecordings" would correspond to a directory above the application.

*FIMrecorder* will fall back to hard-coded defaults and create a new configuration file if you happen to delete it.

## 4.2 .pfs Files

In addition to `settings.json` there are `.pfs` files in your **Configuration Directory** for every camera model you've used in *FIMrecorder*. Those text files are being generated when you use a device for the first time with *FIMrecorder* and include all the parameters of the specific model.

Those files can be used to modify the resolution, offset and binning parameters of your device. If in doubt, take a look at your **Default Camera Parameters** for a comparison. It is recommended to not change any other parameters in these files unless you've read the documentation for your camera model provided by Basler.

## 5 Basic Workflow

The following sections describe the basic steps to use *FIMrecorder* for a successful measurement.

### 5.1 Pre-Recording

Most of the work goes into steps preceding the actual measurement. Therefore, quite a bit attention to details is required here.

#### 5.1.1 Checking Setup

1. Prepare your FIM setup (agarose gel, barriers, everything that you don't want to be tracked).
2. Turn on the LEDs of your FIM setup.

---

<sup>6</sup><https://fourcc.org/codecs.php>

<sup>7</sup>[https://docs.opencv.org/4.0.1/d7/d9e/tutorial\\_video\\_write.html](https://docs.opencv.org/4.0.1/d7/d9e/tutorial_video_write.html)

3. Make sure your camera is connected to a USB3 port with power supply.
  - If your camera was disconnected on starting *FIMrecorder*, simply connect your camera and restart the application.
4. Start *FIMrecorder*.
5. If necessary, adjust focus, aperture and field of view manually on your camera.
  - Use the lens feature (11) to make focussing easier.
  - If background subtraction is already enabled, disabling it might make this step easier.
6. Depending on the surrounding environment, it might be beneficial to dim or turn off lights.
  - This is often not necessary but improves image acquisition a lot.

### 5.1.2 Measurement Annotations

It is recommended to fill in every field in the “Measurement” tab (7). However, only setting the **Recording Duration** is mandatory. A timestamp will be set automatically. This data will be stored in a `.json` file alongside the recorded footage (see Locating Recorded Data).

### 5.1.3 Applying Camera parameters

The following settings are applied in hardware before each single frame is acquired:

**Exposure Time:** This value determines how long the sensor is exposed to light for every frame.

Enabling **Auto Exposure** might help to determine a good range but it is not recommended for the actual measurement.

The LEDs of your FIM setup have a characteristic flickering frequency which reduces the quality of your image signal. To mitigate this effect, it is necessary to find an exposure time which is long enough to capture at least *one flicker*.

E.g. on my tested setup values of 20000 or 30000 worked quite well. This might differ for your setup.

default: 20000

**Frame Rate:** The frame rate determines how many frames are captured in 1 second.

For FIM purposes the default value should be sufficient in most cases.

Note that this setting also interacts with the LEDs’ characteristic flickering frequency. Also,  $\frac{1}{\text{Frame Rate}}$  should be greater than or equal to **Exposure Time**.

default: 20.0

**Gamma Correction:** Roughly speaking, this setting adjusts the brightness of the image signal.

$$\text{new pixel value} = \left( \frac{\text{old pixel value}}{255} \right)^\gamma \cdot 255$$
 where pixel values range between 0 and 255.

default: 1.0

**Gain:** Roughly speaking, this setting adjusts the apparent brightness of the image signal or how strong the sensors reaction to light is.

default: 0.0

**Black Level:** This setting determines the lowest possible value of a pixel, e.g. a black level of 20 applies an offset of 20 to each pixel value.

default: 0

### 5.1.4 Real-Time Signal Modifications

The following settings are applied in-software after each frame is acquired. The order of operation corresponds to the order in the UI from top to bottom.

**Subtract Background:** This will accumulate a certain number of frames which are being used to build a static background image which will be subtracted from every acquired frame.

Since the background image is static, **it is necessary to recalculate the background image when changing camera parameters (See Applying Camera parameters).**

Disabling this setting does not delete the current background image. This way you can compare the effect by toggling the checkbox.

The time it takes to acquire the background image depends on the **Frame Rate**.

As a rule of thumb, **place every object you want to be tracked on your FIM setup *after* calculating the background image.**

default: 100

**Cutoff Threshold:** This sets every pixel value below the specified threshold to 0 (i.e. black).

default: 0

**Stretch Histogram:** This setting basically multiplies pixel values by themselves with a certain *stretching* factor.

new pixel value =  $\frac{\text{stretching factor}}{255} \cdot (\text{old pixel value})^2$  where pixel values range between 0 and 255<sup>8</sup>.

default: 0.0

## 5.2 Recording

Make sure to not cause any strong vibrations in your FIM setup since some settings are sensitive to such disturbances (e.g. **Subtract Background**). Other than that, you just have to wait. The progress bar (12) displays the elapsed time.

## 5.3 Post-Recording

To use acquired data for further analysis, the following information could be useful.

---

<sup>8</sup>Resulting values above 255 will be clipped to 255.

### 5.3.1 Locating Recorded Data

Recorded video files can be found in subdirectories (named after time and date of the recording) of **Recording Directory**. The video files themselves also consist of time and date suffixed by **Video Container Format** (i.e. ".avi" by default). Measurement-related metadata can be found in a corresponding **.json** file.

### 5.3.2 Extracting Frames from Video Files

To extract frames from video, simply push the corresponding button (5). This step is necessary in conjunction with FIMTrack, which requires single images as input for tracking larvae. *FIMrecorder* creates a folder named **frames** besides the video file containing all of the extracted frames.

After analysis with FIMTrack, the extracted frames can be deleted as they occupy quite a lot of disk space. The video files are suited better for archiving purposes.

## 6 Troubleshooting

Feel free to open an issue on github<sup>9</sup> if you have any trouble. Please make sure to include a recent debug log.

### 6.1 Known Limitations

Besides the limited set of supported devices, there are a few more limitations to this date:

- **Only an aspect ratio of 1:1 is displayed correctly in the preview.**  
This does not affect recording.
- Changing resolution requires editing **.pfs** files
- Only **Mono8** pixel formats are supported
- Only the default codec and container format are supported

### 6.2 Logging

*FIMrecorder* generates **debug.log** files which may help track down possible culprits if you encounter any problems. Please make sure to copy your debug logs as soon as you encounter a problem. Otherwise the important information will be overwritten.

Older debug logs may still be helpful and are suffixed by ascending digits (e.g. **debug.log.1**).

---

<sup>9</sup><https://github.com/fncnt/fimrecorder/issues/new>

### 6.2.1 `loggingconf.json`

To change the logging behaviour, `loggingconf.json` in your **Configuration Directory** can be edited. In most cases, you shouldn't be required to do so.

See `logging.config`<sup>10</sup> for further information.

---

<sup>10</sup><https://docs.python.org/3.6/howto/logging-cookbook.html#an-example-dictionary-based-configuration>