

---

# **Pd-Motion Documentation**

***Release 1.0.3***

**Matthieu Berjon**

**Nov 01, 2017**



---

## Contents:

---

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Important note . . . . .	3
1.2	Dependencies . . . . .	3
1.3	Setup of the abstractions . . . . .	3
1.4	Compilation of the external . . . . .	4
1.5	Contributing . . . . .	4
1.6	Submitting bugs . . . . .	5
<b>2</b>	<b>Abstraction and externals documentation</b>	<b>7</b>
2.1	Abstractions . . . . .	7
2.2	Externals . . . . .	8



Pd-motion (formerly MbLib) is a set of abstractions for [Pure Data](#) that helps to manipulate the spatialisation of audio streams and handle head tracking data coming from a Flock of Bird.

This project has been developed within the Action and Perception group of the school of Psychology at Cardiff University. A new audiovisual lab has been created and needed a bit of code in order to be able to use properly the equipments. Pure Data has been chosen for several reasons:

- able to drive an old Flock of Bird head tracker out of the box
- Real time capacities
- Run on Windows, GNU/Linux and MacOSX
- Open source software

You'll find several abstractions to manipulate and acquire data from the head tracker, basic filtering, spatialisation, a staircase method and so on. All the objects are supposed to be as generic as possible and should be usable outside of its main purpose in the lab.



I invite you to use PdExtended rather than the vanilla version of Pure Data. For the compilation it won't have any impact (as it uses the same library `pdlib.h`) but at least you'll have all the integrated other libraries on which some abstractions are relying on.

### 1.1 Important note

At the moment the current makefile does not work properly and will be corrected really soon.

### 1.2 Dependencies

In order to work properly, the library relies on several other 3rd party objects. In order to avoid any issues it is recommended to install the extended version of Pure Data called [Purr-Data](#). If you use the Vanilla version, you will need to install the following libraries before installing this one.

- [IEMLib](#)
- [Cyclone](#)
- [list-abs](#)
- [bsaylor](#)
- [cxc](#)
- [comport](#)

### 1.3 Setup of the abstractions

At the moment, the abstractions doesn't come in a handy library. In order to set up the system, you will need to give Pure Data the path to the abs folder in *file/preferences/path*.

## 1.4 Compilation of the external

### 1.4.1 Linux

In order to compile and install this library you will need:

- gcc >= 4.7.1
- gnu-make >= 3.82
- Pure Data >= 0.45

To compile the library:

```
make make install
```

All file will be added to you default Pure Data external folder ~/pdextended.

### 1.4.2 Windows

In order to compile under windows, you will need:

- Mingw
- Pure Data >= 0.45

To compile the library:

```
mingw32-make mingw32-make install
```

All files will be install for the current user in %%application\_data%%

### 1.4.3 MacOS

The Makefile does'nt take into account MacOS compilation at the moment.

Installation of abstraction License ——

Please refer to the LICENSE file at the root of the project.

## 1.5 Contributing

We're really happy to accept contributions from the community, that's the main reason why we open-sourced it! There are many ways to contribute, even if you're not a technical person.

We're using the infamous [simplified Github workflow](#) to accept modifications (even internally), basically you'll have to:

- create an issue related to the problem you want to fix (good for traceability and cross-reference)
- fork the repository
- create a branch (optionally with the reference to the issue in the name)
- hack hack hack
- commit incrementally with readable and detailed commit messages
- submit a pull-request against the master branch of this repository



We'll take care of tagging your issue with the appropriated labels and answer within a week (hopefully less!) to the problem you encounter.

If you're not familiar with open-source workflows or our set of technologies, do not hesitate to ask for help! We can mentor you or propose good first bugs (as labeled in our issues). Also welcome to add your name to Credits section of this document.

## 1.6 Submitting bugs

You can report issues the issue tracker of the [project](#), that would be a really useful contribution given that we lack some user testing on the project. Please document as much as possible the steps to reproduce your problem (even better with screenshots).



---

## Abstraction and externals documentation

---

### 2.1 Abstractions

#### 2.1.1 Audio

**audio-stim** Generate audio stimulus properties.

**bandpass~** Band-pass filter with a bypass option.

**bandpass24~** 24 channels band-pass filter with a bypass option. (Requires bandpass~).

**fadeio~** Fade in/out manager.

**gaussian-gain** Compute the gain to apply to a speaker depending of the source size.

**gaussian-spat24** Compute the gains to apply to a 24 channels sound system according to the position of the speakers and the source width. (Requires gaussian-gain).

**master24~** 24 channels master with a fade in/out manager. (Requires fadeio).

**motion24~** Spatialize an incoherent stimulus (noise for example) over 24 channels. (Requires gaussian-spat24).

**mute24~** 24 audio channels mute.

**noise24~** 24 audio channels of white noise. Each channel generate independent noise (useful to simulate source motion).

**partconv-im~** 1 channel filtering object using partitionned convolution method. This object can be bypass. (Requires partconv~).

**playfile~** Play any wave file in argument using a table method.

**spectral-corr24~** 24 audio channels applying spectral filtering on input signals. This object can be bypass if necessary. (Requires partconv-im~).

**stimulus-position** Generate the azimuth position stream according to several parameters.

**vumeter~** GUI object displaying the level in dBfs (full scale) of an audio signal.

### 2.1.2 Head tracking (Flock of Bird)

**bird-conv-ang** Conversion of an angular Flock of Bird raw data flow in azimuth, elevation and roll in degrees.

**bird-conv-pos** Conversion of a position Flock of Bird raw data flow in x, y and z in inches or centimeters.

**bird-ctl** Flock of Bird acquisition control features.

**bird-raw-pos-ang** Flock of Bird raw flow extraction. Output two separate raw flows, one for position, one for angles.

### 2.1.3 Miscellaneous

**cmp-gate** Comparative gate using 2 boundaries.

**get-button** Send a unique bang when the right key is hit.

**get-displacement** Get the displacement according to the duration and speed of an object.

**in2cm** Convert inches in centimeters.

**percentage** Compute a given percentage of a value.

**pr-bit** Generate a pseudo random bit (0 or 1) and then alternate the value.

**rand** A superior layer of random object. You just to specify a range of value from 0 and shift of this range.

**serial-ctl** Superior layer to comport in order to open/close, specify a baud rate for the serial port.

## 2.2 Externals

**staircase** Staircase method.