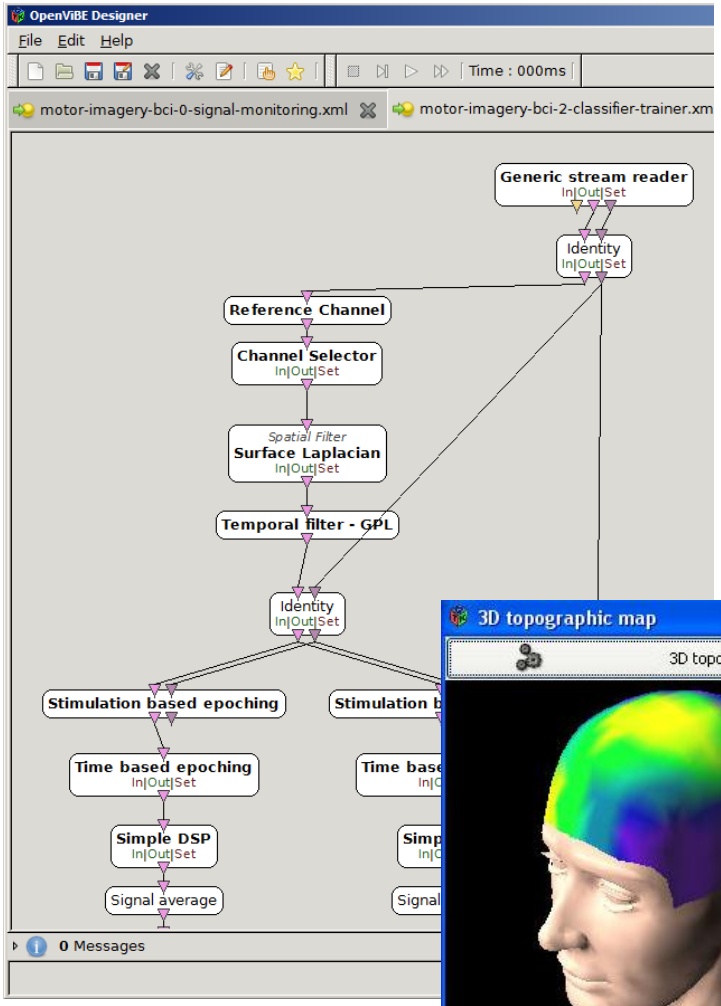
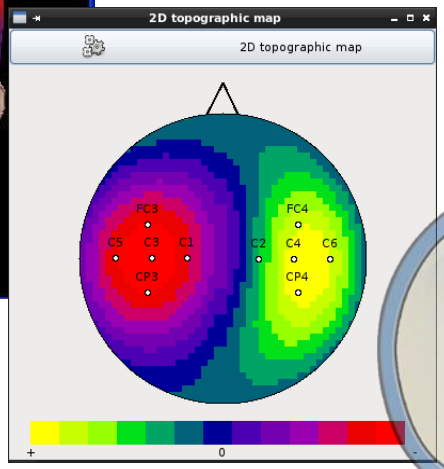
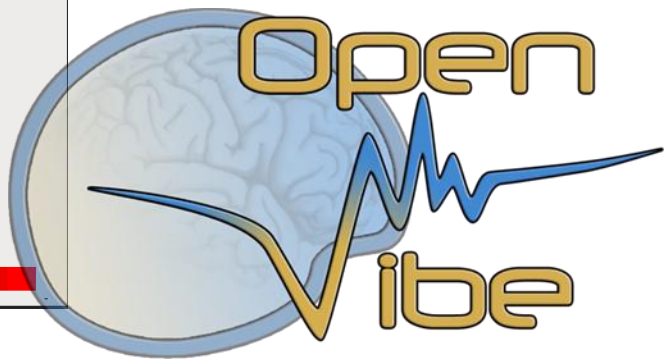
Introduction to OpenViBE

Jussi T. Lindgren, PhD
jussi-tapio.lindgren@inria.fr
HYBRID / Inria Rennes







Part I
What is OpenViBE?

Part II
Interoperability

An open source software platform for BCI

Enjoyed by research labs, clinicians, teachers, game developers and hobbyists worldwide

OpenViBE Team 2013

Coordination at Inria (group HYBRID @ Rennes, France)

- http://team.inria.fr/hybrid/
- Primary research area: Virtual Reality
- Project leader: Anatole Lécuyer
- OpenViBE lead engineers: Jozef Legény & Jussi T. Lindgren

Landscape at Inria:

- Involved teams : HYBRID (Rennes), ATHENA (Sophia), POTIOC (Bordeaux), NEUROSYS (Nancy)
- Involved personnel: 5 researchers & 5 developers

Industry:

- Earlier OpenViBE developers Yann Renard & Laurent Bonnet now work at a related startup

Key features of OpenViBE

- Data acquisition from various biosignal devices
- Graphical design of signal processing chains
- Plugins for DSP, classification, visualization and more
- Online and offline data processing
- Core in C++. Interfaces with script languages and various file formats.

Architecture

- Boxes are plugins
- Solid arrows denote EBML-encoded streams of different types (signal, stimulation, ...)

Arbitrary DAG designs are possible
(Shown chain is for illustration only)

Server: Its mostly about drivers

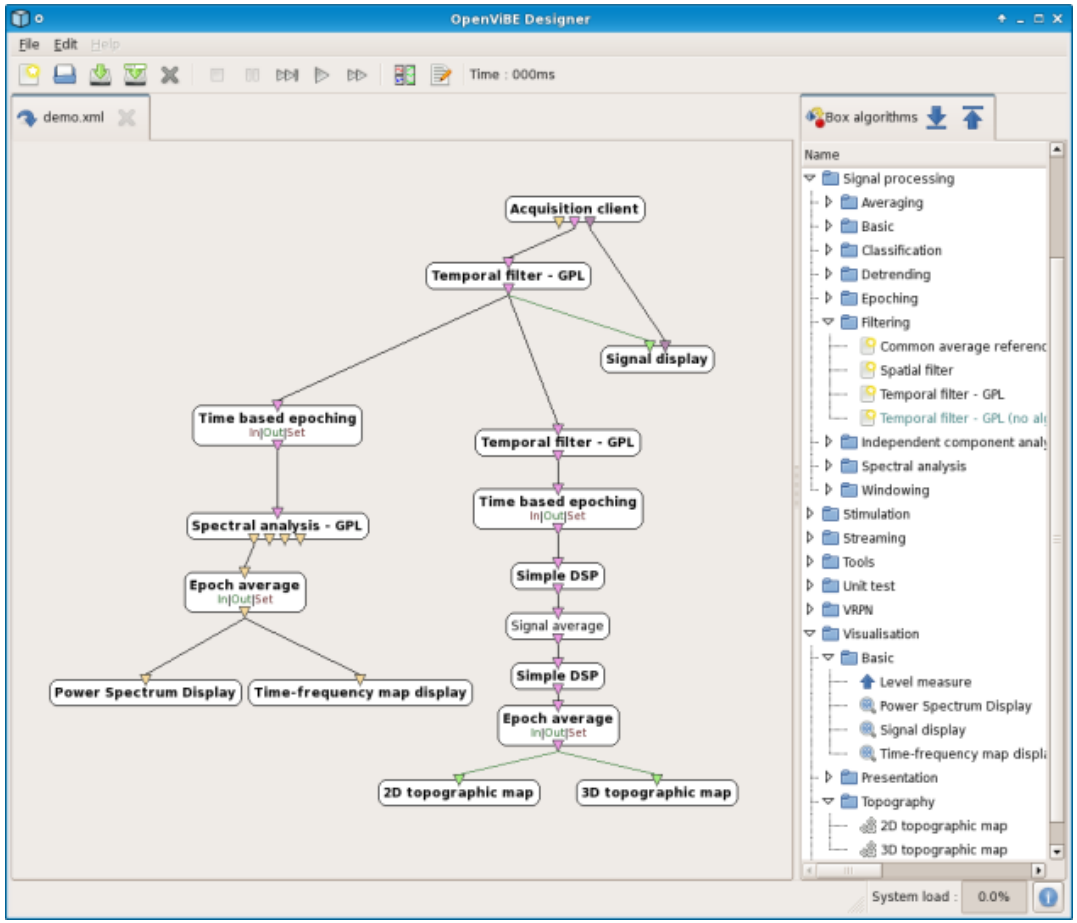
Vendor	Devices
BrainMaster	Atlantis, Discovery
Brain Products	actiCHamp, BrainAmp, ...
Emotiv	EPOC
g.Tec	gUSBamp, gMobilab+
MicroMed	SD LTM
MindMedia	NeXus32B
NeuroSky	MindSet (MindWave in SVN)
OpenEEG	MonolithEEG, Modular EEG P2
TMSi	Porti32

+ more unstable, unsupported & beta drivers

+ drift correction, software tagging, …

“Designer” Client

Make your own DSP chains



No programming skills required

Client plugins include

DSP

Basic math, detrending, epoching, spatial and temporal filters, spectral analysis, signal generators, statistics, FastICA, windowing, ...

Visualizations

Signal, Power Spectrum, Graz, Matrix, Topographic Maps: 2D & 3D, Cue Image, Voxel Display, ...

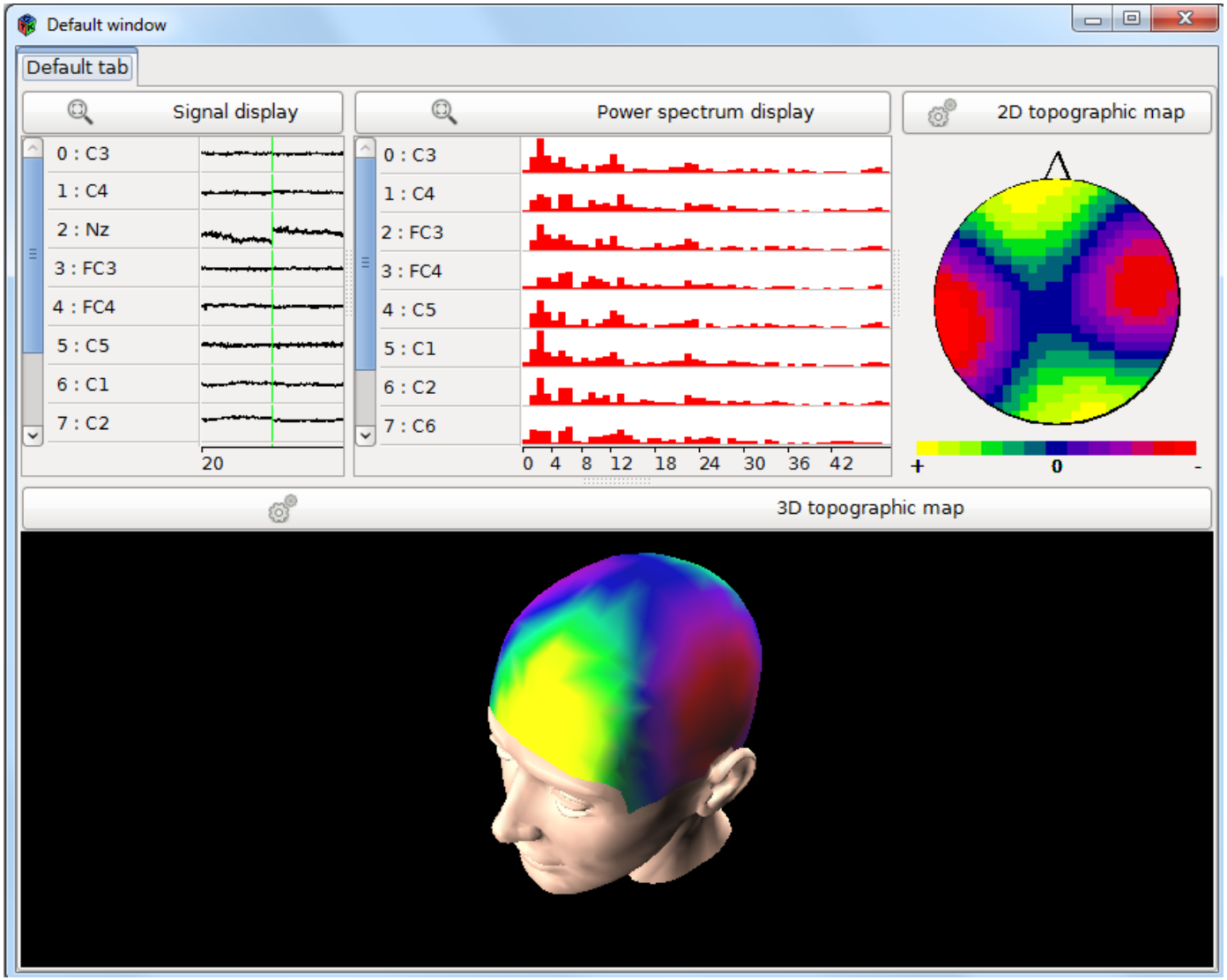
Classifiers

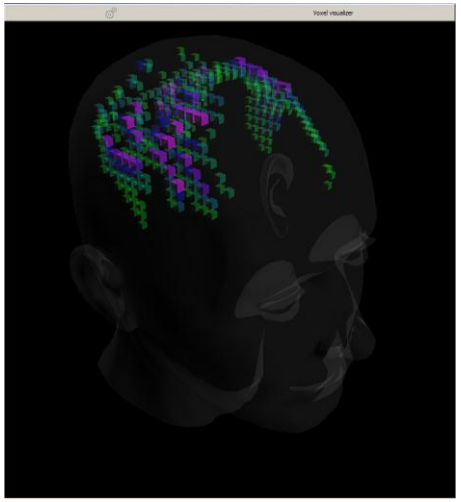
SVM, LDA

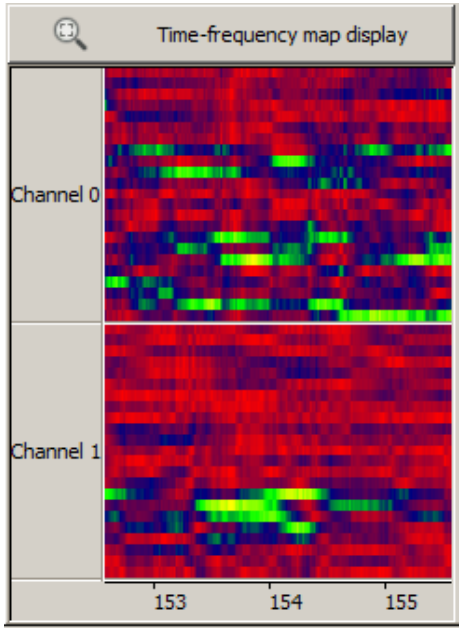
Other

Script language interfaces, file I/O, etc ...

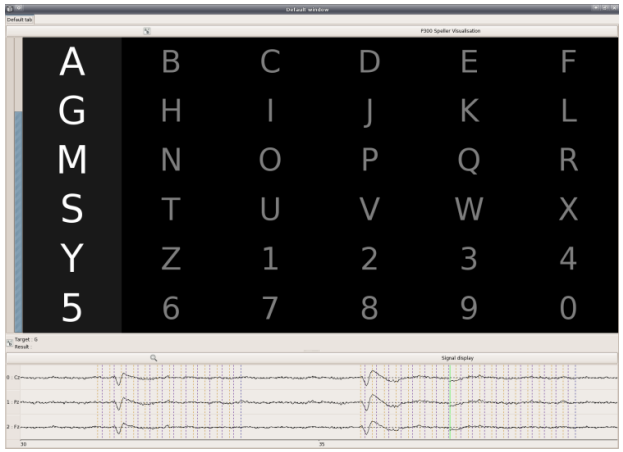
Client: Various real-time displays



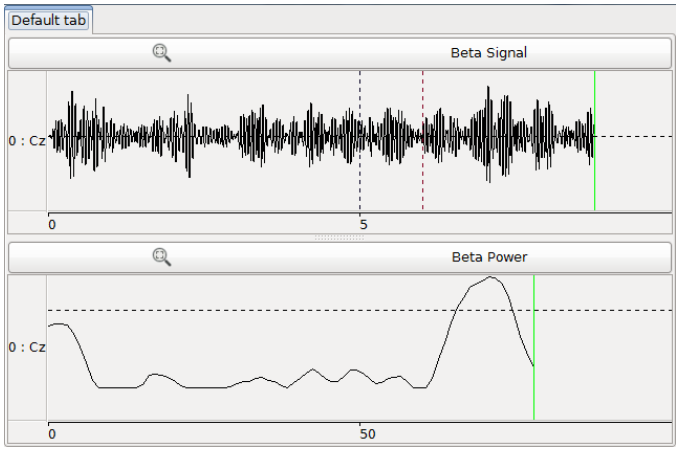




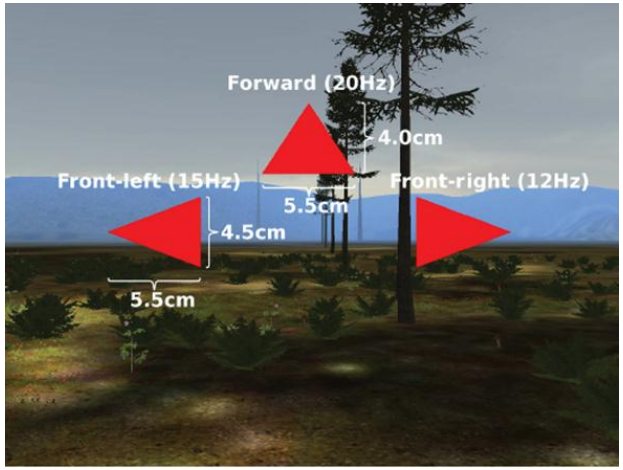
Some approaches implemented on OpenViBE



P300



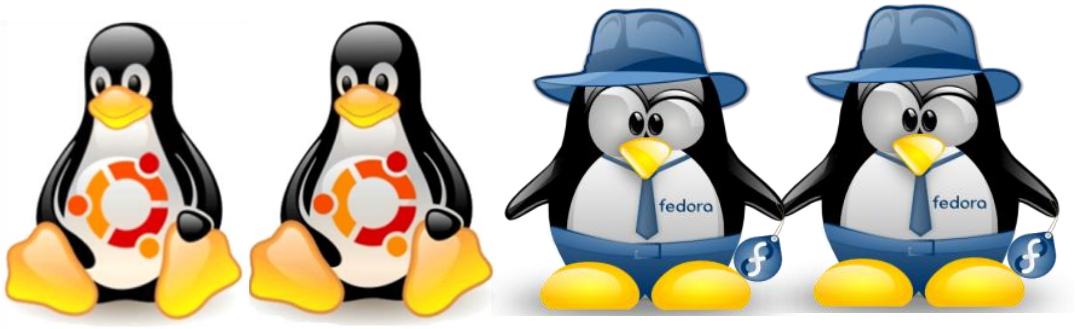
Neurofeedback



SSVEP

Motor Imagery

OS’es & dev-tools supported



Ubuntu: LTS & latest. Fedora: 2 latest

Linux: source only

GNU C++ build environment required

Windows XP & Windows 7

Windows: binary available

or compile with MSVC

In SVN: Visual Studio IDE support

(release in June-July 2013)

Software requirements

- Mandatory dependencies
  - Server: Cmake, Boost, GTK
  - Client: Cmake, Boost, Ogre, OIS, CEGUI, GTK, Expat
- Strong recommendations
  - Client: VRPN, ITPP
- Optional
  - Client: ALUT, Eigen3, Lua, Matlab, Python27, Vorbis

- On Linux, most dependencies are used from distribution packages or downloaded by the installer
- On Windows, the installer downloads the deps

License

v0.15.0 and before:

Codebase mostly under LGPL 2

Some plugins under GPL2

Near future:

The whole codebase will likely switch to

Want to see your code in the OpenViBE distribution?

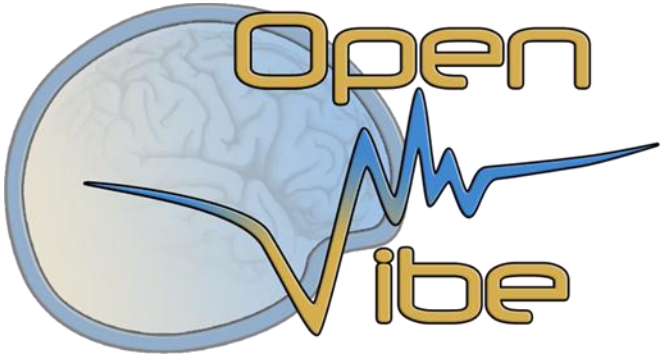
- New stuff : New plugins, drivers, scenarios and applications can be freely contributed. You keep your rights.
- Changes to core components: Rights transfer to Inria required

Commitment to OpenViBE

- Past development effort: ≈ 22 man-years at Inria (and various multi-partner projects since 2005)
- On-going support (2013-2015): 4 Inria engineers with 100% allocation to OpenViBE
- Inria startup: Mensia Technologies (launched in Nov 2012) can support commercial users of OpenViBE

More information at

http://openvibe.inria.fr



Part I
What is OpenViBE?

Part II
Interoperability

Current I/O plugins in OpenViBE

File I/O

Format	Read	Write
BCI2000	X	
BCI comp. IIIb	ASCII	
Brainamp	X	
CSV	X	X
EDF		X
GDF	X	X
OV (≈ EBML)	X	X
Raw	X	

Direct I/O

Type	Read	Write
Lua	Stim	Stim
Matlab	X	X
Python	X	X
TCP/IP	EBML	
VRPN	X	X

Understanding OpenViBE

Well-defined kernel interfaces and stream specifications exist

Good to know…

- OpenViBE uses 32:32 fixed point simulated time, in seconds
- OpenViBE .ov files are catenated EBML chunks w/o headers
- There is no SDK: build against the OpenViBE source tarball

User and developer documentation gets you started

http://openvibe.inria.fr/documentation-index/

Making OpenViBE understand ‘XYZ’

Plugin contributions are very welcome

E.g.

- Readers/writers for new file formats
- Language plugins (like Matlab & Python boxes)
- Boxes to pass data to/from external software
- Drivers that actually read from another application
  - TiA plugin already exists as an extension from Gipsa-Lab

We’re happy to collaborate on developments

Catch us by email, forum, or IRC

http://openvibe.inria.fr/contact/