METISS Team

The Matching Pursuit Tool Kit

Getting Started

Authors:
Sacha Krstulović
Rémi Gribonval

Contributors:
Ronan Le Boulch
Benjamin Roy

March 1, 2012

Contents

1. Introduction

1.1 Learning about MPTK

The Matching Pursuit Tool Kit (MPTK) provides a fast implementation of the Matching Pursuit algorithm for the sparse decomposition of multichannel signals such as audio signals. It comprises a library, standalone utilities and Matlab scripts.

MPTK provides an implementation of Matching Pursuit which is:

- **FAST:** e.g., extract 1.5 million atoms from a 1 hour long, 16kHz audio signal (15dB extracted) in 0.25x real time on a Pentium IV@2.4GHz, out of a dictionary of 178M Gabor atoms. Such incredible speed makes it possible to process "real world" signals.
- **FLEXIBLE:** multi-channel, various signal input formats, flexible syntax to describe the dictionaries ↦ reproducible results, cleaner experiments.
- OPEN: modular architecture ↦ add your own atoms ! Free distribution under the GPL.

MPTK is mainly developed and maintained within the METISS Research Group (http://www.irisa.fr/metiss/) on audio signal processing, at the INRIA Research Institute (http://www.irisa.fr or http://www.inria.fr/rennes) in Rennes, France.

1.2 Reading this document

This document describes the basic principles about how to download, install and use the Matching Pursuit Tool Kit. It is divided into two major sections :

- Downloading and installing MPTK depending on the system OS
- Understanding the basic usage of MPTK with command line tools
- Understanding the basic usage of MPTK within Matlab

If you need more specific details about any part of this Getting Started document, please refer to the following chapters of the **User manual** document :

- **Chapter ??** for how to pre-build, build and install MPTK from the source files
- **Chapter ?? & ??** for how to use MPTK with command line tools
- **Chapter ?? & ??** for how to use MPTK within Matlab

The User manual document can be downloaded here :

- https://gforge.inria.fr/docman/?group_id=36

Part I

Installing MPTK

2. MPTK for Windows

2.1 Downloading MPTK

The latest version of MPTK is available at (https://gforge.inria.fr/frs/?group_id=36). Depending on the processor architecture of your computer, you will have to download either the 32 bits package or the 64 bits package:

- For Windows 32 bits : "MPTK-binary-v.v.v-i386-Windows.exe"
- For Windows 64 bits : "MPTK-binary-v.v.v-x86_64-Windows.exe"

<u>Hint :</u> *To find the processor architecture of your computer:*

- *Open a terminal command using :*
  - *Start ↦ All Programs ↦ Accessories ↦ Command Prompt*
- *Use the following command :* echo %PROCESSOR_ARCHITECTURE%
  - *If the answer is "x86" then your OS is 32 bits*
  - *If the answer is "AMD64" then your OS is 64 bits*

2.2 Installing MPTK

When double clicking the executable "MPTK-binary-v.v.v.-(i386/x86_64)-Windows.exe":

1. Accept the terms of the licence agreement

2. Select the path folder where to install MPTK (we'll call it *"path_to_MPTK"*)

   - We suggest to use the default folder : "C:\Program Files\MPTK"

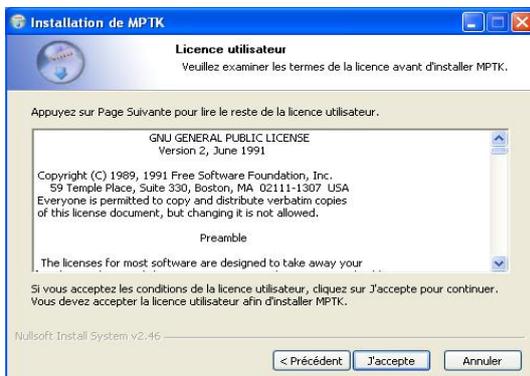3. Finish the installation

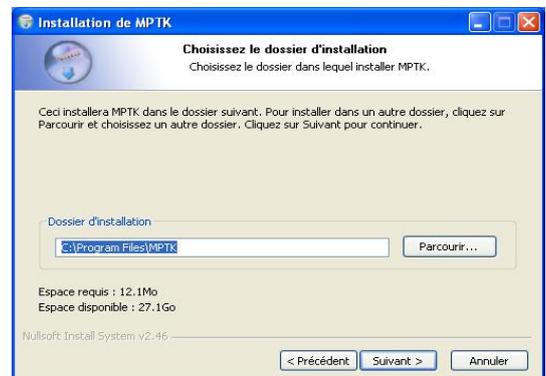Figure 2.1: Licence agreement

Figure 2.2: Path folder selection

2.3 Configuring the path

An environment variable called MPTK_CONFIG_FILENAME needs to be set, either temporarily, or permanently, with the path of the "path.xml" file located in the *"path_to_MPTK/mptk"* directory. This file defines the environment paths that MPTK needs to work correctly.

2.3.1 Temporary path configuration

Here is the way to temporarily configure the MPTK_CONFIG_FILENAME environment variable. **Warning** : This is a temporary setting and it needs to be done at each reset of the computer.

- Open a terminal command using :
  - Start ↦ All Programs ↦ Accessories ↦ Command Prompt
- Use the command : *set MPTK_CONFIG_FILENAME = path_to_MPTK/mptk/path.xml*
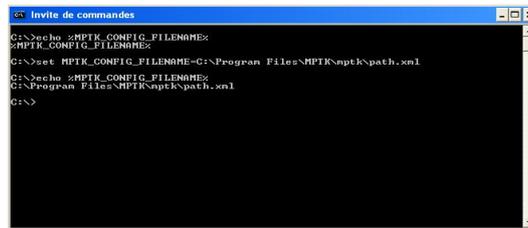
Figure 2.3: Filled command prompt

2.3.2 Permanent path configuration

Here is the way to permanently configure the MPTK_CONFIG_FILENAME environment variable

- Check if the environment variable is correctly set with : echo %MPTK_CONFIG_FILENAME%
- Open the environment variable configuration panel situated under :
  - Start ↦ Config panel ↦ System ↦ Advanced ↦ Environment variables
- Add a new user variable with :
  - Name : MPTK_CONFIG_FILENAME
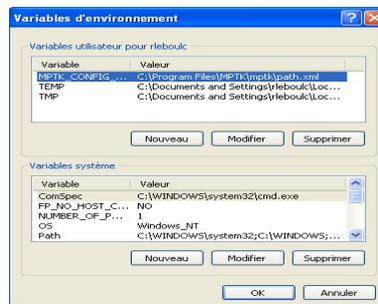 - Value : *path_to_MPTK/mptk/path.xml*

Figure 2.4: Environment variable configuration

2.3.3   Matlab path configuration

When launching Matlab, the user needs to configure Matlab to work with MPTK:

- Configure the working path either by:
  - Selecting the current folder as "path_to_MPTK/mptk/matlab"
  - Adding the working path using addpath("path_to_MPTK/mptk/matlab")

3. MPTK for Linux

3.1 Downloading MPTK

The latest version of MPTK is available at (https://gforge.inria.fr/frs/?group_id=36). Depending on the processor architecture of your computer, download either the 32 bits package or the 64 bits package:

- For RedHat, Suse, Fedora, Mandriva
    - 32 bits : "MPTK-binary-v.v.v-i386-Linux.rpm"
    - 64 bits : "MPTK-binary-v.v.v-x86_64-Linux.rpm"
- For Debian, Knoppix, Ubuntu
    - 32 bits : "MPTK-binary-v.v.v-i386-Linux.deb"
    - 64 bits : "MPTK-binary-v.v.v-x86_64-Linux.deb"

Hint : To find the processor architecture of your computer :

- *Open a terminal command and use the following command :* **uname -m**
    - *If the answer is "i386" then your OS is 32 bits*
    - *If the answer is "x86_64" then your OS is 64 bits*

3.2 Obtaining additional required packages

Two additional packages are needed. Their installation require administrator privileges on the machine. The "sudo" command may ask you to input administrator password :

- Libsndfile (tested with version 1.0.23) pre compiled library
- FFTW (tested with version 3.2.2) pre compiled library

You can see below some examples about how to download those libraries using terminal :

Ubuntu	Fedora	Mandriva
sudo apt-get install -y -qq libsndfile1-dev	sudo yum -y -qq install libsndfile-devel	sudo smart install -y -qq libsndfile1-dev
sudo apt-get install -y -qq libfftw3-dev	sudo yum -y -qq install fftw-devel	sudo smart install -y -qq libfftw3-dev
	sudo yum -y -qq install fftw-static	

3.3 Installing MPTK

Depending on the type of Linux you have there ares two ways to install the packages :

- for "rpm" package : *rpm -ivh MPTK-binary-v.v.v-(i386/x86_64)-Linux.rpm*
- for "deb" package : *dpkg -i MPTK-binary-v.v.v-(i386/x86_64)-Linux.deb*

3.4 Configuring the path

An environment variable called MPTK_CONFIG_FILENAME needs to be set, either temporarily, either permanently, with the path of the "path.xml" file located in the *"path_to_MPTK/mptk"* directory. This file defines the environment paths that MPTK needs to work properly.

3.4.1 Temporary path configuration

Here is the way to temporarily configure the MPTK_CONFIG_FILENAME environment variable. **Warning :** This is a temporary setting and it needs to be done at each reset of the computer.

- *With Bash shell :*
    - *export MPTK_CONFIG_FILENAME ="path_to_MPTK/mptk/path.xml"*
- *With C-shell :*
    - *setenv MPTK_CONFIG_FILENAME "path_to_MPTK/mptk/path.xml"*
- *You can check if the environment variable is correctly set with :*
    - *echo $MPTK_CONFIG_FILENAME*

3.4.2 Permanent path configuration

In order to permanently configure the MPTK_CONFIG_FILENAME environment variable, add the bash shell (or the C-shell) configuration line to the ".bashrc" (or the ".cshrc") file situated under your home directory.

3.4.3 Matlab path configuration

When launching Matlab, the user needs to configure Matlab to work with MPTK:

- Configure the working path either by:
    - Selecting the current folder as *"path_to_MPTK/mptk/matlab"*
    - Adding the working path using *addpath("path_to_MPTK/mptk/matlab")*

4. MPTK for Mac OS

4.1 Downloading MPTK

The latest version of MPTK is available at (https://gforge.inria.fr/frs/?group_id=36). Depending on the processor architecture of your computer, you will have to download either the 32 bits package or the 64 bits package:

- For Mac 32 bits : "MPTK-binary-v.v.v-i386-Mac.exe"
- For Mac 64 bits : 'MPTK-binary-v.v.v-x86_64-Mac.exe"

Hint : _To find the processor architecture of your computer :_

- _Open a terminal command and use the following command :_ **_uname -m_**
    - _If the answer is "i386" then your OS is 32 bits_
    - _If the answer is "x86_64" then your OS is 64 bits_

4.2 Obtaining additional required packages

Two additional packages are needed. Their installation require administrator privileges on the machine. The "sudo" command may ask you to input administrator password :

- Libsndfile (tested with version 1.0.23) pre compiled library
- FFTW (tested with version 3.2.2) pre compiled library

You can see below some examples about how to download those libraries using terminal:

Mac
sudo /opt/local/bin/port install libsndfile +universal
sudo /opt/local/bin/port install fftw-3 +universal

Hint : _We suggest to use the "port" command from MacPorts because the command "+universal" allows to retrieve libraries which are compatible with both system architectures (32 bits and 64 bits). The package is available at http://www.macports.org/install.php_

4.3 Installing MPTK

When double clicking the executable **"MPTK-binary-v.v.v.-(i386/x86_64)-Mac.dmg"**:

1. Accept the terms of the licence agreements

2. Accept the path folder where to install MPTK

    - The default and unique folder is : "/usr/local/"
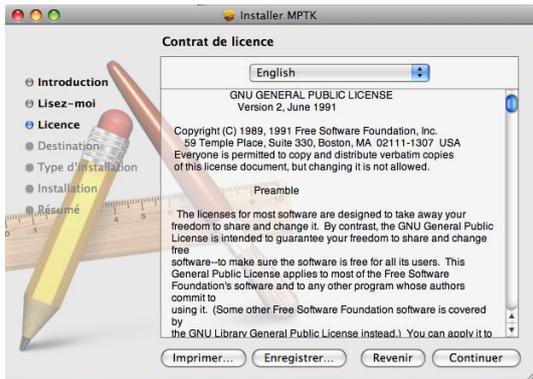
3. Finish the installation
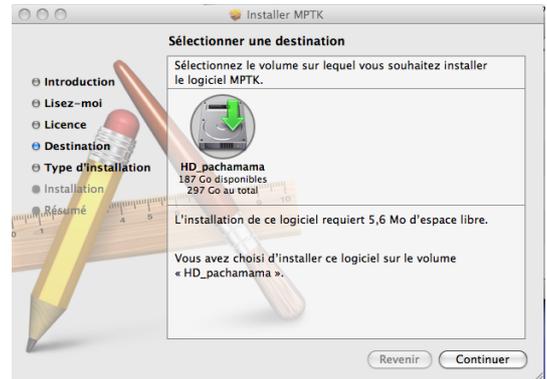
Figure 4.1: Licence agreement

Figure 4.2: Path folder selection

4.4 Configuring the path

An environment variable called MPTK_CONFIG_FILENAME needs to be set, either temporary, either permanently, with the path of the path.xml file located in the *"path_to_MPTK/mptk"* directory. This file defines the environment paths that MPTK needs to work properly.

4.4.1 Temporary path configuration

Here is the way to temporary configure the MPTK_CONFIG_FILENAME environment variable. **Warning :** This is a temporary setting and it needs to be done at each reset of the computer.

- *With Bash shell :*
  - export MPTK_CONFIG_FILENAME ="/usr/local/mptk/path.xml"
- *With C-shell :*
  - setenv MPTK_CONFIG_FILENAME "/usr/local/mptk/path.xml"
- *You can check if the environment variable is correctly set with :*
  - echo $MPTK_CONFIG_FILENAME

4.4.2 Permanent path configuration

In order to permanently configure the MPTK_CONFIG_FILENAME environment variable, add the bash shell (or the C-shell) configuration sentence to the ".bashrc" (or the ".cshrc") file situated under your home directory.

4.4.3 Matlab path configuration

When launching Matlab, the user needs to configure Matlab to work with MPTK:

- Configure the working path either by :
  - Selecting the current folder as "/usr/local/mptk/matlab"
  - Adding the working path using addpath("/usr/local/mptk/matlab")

Part II

Using MPTK

5. MPTK from within Matlab

In the following, we assume that MPTK has been installed correctly, that the system path configuration has been set and that Matlab is installed.

5.1 Getting Started

GettingStarted command is used for a better understanding of MPTK functionalities. This script is divided in two parts. The first part consists in retrieving the environment informations and the available plugins. The second part consists in describing several tutorials about MPTK utilities.

1st part description : Example of available plugins and dictionaries
Here is the list of types of atoms available in MPTK plugins:

- anywave	- anywavehilbert
- constant	- dirac
- gabor	- harmonic
- mclt	- mdct
- mdst	- nyquist

As well as information on the path where reference files can be found:

- path_to_MPTK/mptk/reference

which can be used to find examples of dictionaries:

dic_anywave.xml	dic_constant.xml	dic_harmonic.xml
dic_mdst.xml	dic_anywave_modifie.xml	dic_dirac.xml
dic_mclt.xml	dic_nyquist.xml	dic_chirp.xml
dic_gabor_two_scales.xml	dic_mdct_two_scales.xml	dic_test.xml

2nd part description : Description of the tutorials
There are several tutorials on using MPTK4Matlab:

1. Dictionaries
2. Books
3. Running Matching Pursuit Toolkit
4. Multichannel decompositions (in preparation)
5. Anywave atoms (in preparation)
6. Demixing pursuit (in preparation)

Dictionaries : How to read (*dictread*), create (*dictwrite*) a dictionary description
Books : What is a book (storage format for sparse signal representations) and how to read (*bookread*), save (*bookwrite*) or plot any book (*bookplot*, *bookover*)
Running MPTK : Procedure to follow if you want to decompose a signal :

- Read a signal (*sigread*)
- Read a dictionary (*dictread*)
- Decompose the signal (*mpdecomp*)

5.2   Getting the environment information

getmptkinfo command is launched under the GettingStarted.m script, and is used to retrieve the environment datas, such as :

- The plugins atoms names available
- The block parameters needed to correctly define each atom
- The windows names that can be used to through the signal waveform
- The example or default environment paths

↦ mptkInfo = getmptkinfo();

5.3   Reading a signal

sigread command reads an imports signal "exampleSignal" of any format supported by lib-sndfile library to Matlab and gives a matrix "signal" (numSamples x numChans) and the sampling frequency of the read signal "sampleRate".

↦ [signal sampleRate] = sigread(mptkInfo.path.exampleSignal);

5.4   Reading a dictionary

dictread command imports a dictionary description "defaultDict" to Matlab and gives a dictionary description with the following structure : dict.block{i} = block where, for example block.type = 'dirac'

↦ dict = dictread(mptkInfo.path.defaultDict);

5.5   Decomposing a signal

mpdecomp command decompose a signal "Signal" using its sampling frequency "sampleR-ate", a dictionary structure "dict" performing "numIter" iterations and gives the resulting decomposition "book", the "residual" obtained after the iterations and "decay", a vector with the energy of the residual after each iteration.

↦ [book residual decay] = mpdecomp(signal,sampleRate,dict,numIter);

5.6   Plotting a book

bookover plots the given "book" over a STFT spectrogram of the given "Signal" for channel "numChan" (or 1 for default). The book and/or the signal can be given as filenames (WAV format for the signal).

↦ figure(5);bookover(book,signal);

5.7 Reconstructing a signal

mprecons *reconstructs the signal from the given "book".*

$\mapsto$ sigrec = mprecons(book);

5.8 Finding further informations on MPTK for Matlab

For further informations on how to use MPTK Matlab functionalities please refer to the Chapter **??** of the User Manual.

6. MPTK command line utilities

6.1   List of commands

Here is a list of the MPTK command line utilities :

- **mpd :** decompose a waveform signal using matching pursuit
- **gpd :** decompose a waveform signal using gradient pursuit
- **mpd_demix :** decompose a waveform signalusing matching pursuit and a mixer matrix
- **mpf :** filters the atoms contained in books
- **mpr :** reconstructs a signal from the atoms contained in a book
- **mpcat :** concatenates any number of books into one
- **mpview :** makes a time-frequency pixmap

6.2   Basic example

Here is a simple example on how to use MPTK with terminal commands :

mpd command iterates Matching Pursuit on signal "sndFileToDecomp.wav" with dictionary "dictFile.xml" and gives the resulting book "bookFile.bin" (and an optional residual signal) after N iterations or after reaching the signal-to-residual ratio SNR.

```
$ mpd -s 10 -R 10 -d dictFile.xml sndFileToDecomp.wav bookFile.bin
```

mpr commands rebuilds a signal "sndReconsFile.wav" from the atoms contained in the book file "bookFile.bin". An optional residual "sndResidFile.wav" can be added.

```
$ mpr bookFile.bin sndReconsFile.wav sndResidFile.wav
```

information : Some book examples are available under "path_to_MPTK/mptk/reference/book"

mpf commands filters the atoms contained in "bookFile.bin" (or stdin), stores those which satisfy the indicated properties in "bookYes.bin" (or stdout) and the others in "bookNo.bin".

```
$ mpf –Freq=[0:110] –len=[0:256] bookFile.bin bookYes.bin bookNo.bin
```

6.3   Finding further informations on MPTK cmd line

For further informations on how to use MPTK command line utilities please refer to the Chapter **??** of the User Manual.

7. Source installation

The source installation archive file (https://gforge.inria.fr/frs/?group_id=36) is available under gforge website . This kind of installation is recommended for :

- Whoever have problems installing the binary files
- Whoever wants to help developing MPTK and/or adding new functionalities to MPTK

For further information on how to pre-build with cmake, build and install MPTK from the source files, please refer to the chapter "Download & install from source" of the Usermanual documentation situated under gforge website (https://gforge.inria.fr/docman/?group_id=36) or under the documentation directory (MPTK UserManual).

8. Help, contact and forums

If you need help with the software:

1. Check if a **more recent release** fixes your problem (https://gforge.inria.fr/frs/?group_id=36)
2. Check if somebody else has had a samilar problem and if a fix exist on the **help forum** (https://gforge.inria.fr/forum/forum.php?forum_id=109)
3. If not, post a message on the **help forum**

If you need documentation about the software:

Some articles exposing scientific results related to MPTK are available in PDF format through the following page (https://gforge.inria.fr/docman/?group_id=36)

If you want specific information :

You can write to us to matching.pursuit@irisa.fr.

Request for help sent to this address won't be answered. Please use the Help forum instead.

Thank you for your interest in The Matching Pursuit ToolKit !