MRI Coil-combine Toolbox

Release 0.2.11

Robbert Harms

Dec 13, 2019

Contents

Chapter 1

Introduction

The MRI Coil-reconstruct Toolbox, MCT, is a small toolbox for combining the channels of a multi-channel MRI acquisition. Where possible, this toolbox uses GPU accelerated routines to speed-up the processing. For example, the weights of the STARC (STAbility-weighted Rf-coil Combination) reconstruction model are fitted using the GPU or using multi-threaded CPU. At the moment MCT supports rSoS (root Sum Of Squares), rCovSoS (same as SoS but than with additional usage of a noise covariance matrix) and STARC.

Beta version notice

Please note that this software is still in beta and that the user interface may change over versions.

1.1 Summary

- features multiple coil combine / reconstruction methods
- command line and python interface
- GPU/multicore-CPU accelerated STARC
- Free Open Source Software: LGPL v3 license
- Python and OpenCL based
- Full documentation: https://mct.readthedocs.io/
- Project home: https://github.com/cbclab/MCT
- Uses the GitLab workflow
- Tags: MRI, coil-reconstruct, image reconstruction, opencl, python

1.2 Data reconstruction

This software contains various reconstruction methods that can be used to combine your channels into one (or more) volumes. Not all reconstruction methods may be applicable to your data, for example the STARC [1] method only works when dealing with fMRI data.

Console

To reconstruct your data using the command line, after installation you can use:

```
$ mct-reconstruct <method> {0..15}.nii
```

Where method at the moment is one of “rSoS”, “rCovSoS” or “STARC”. Some methods require more information to combine the channels, please see the full documentation for this.

If you only want to use certain volumes of your data, use the “–volumes” or “-v” switch on the command line:

```
$ mct-reconstruct <method> {0..15}.nii -v odd
```

To use (for example) only the odd volumes. Available options are “odd”, “even” or a list of indices, such as “0 2 4 5” (space separated).

Python

It is also possible to reconstruct your data using the Python API, for example:

```
from mct.reconstruction_methods import rSoS, rCovSoS, STARC

input_path = '/data/'
output_path = '/data/output/'
nmr_channels = 16
input_filenames = [input_path + str(ind)
                   for ind in range(nmr_channels)]

method = rSoS(input_filenames)
method.reconstruct(output_path, volumes='odd')
```

This would reconstruct your data using rSoS using only the odd volumes.

References:

1) Simple approach to improve time series fMRI stability: STAbility-weighted Rf-coil Combination (STARC), L. Huber et al. ISMRM 2017 abstract #0586.

1.3 Quick installation guide

The basic requirements for MCT are:

- Python 3.x (recommended) or Python 2.7
- OpenCL 1.2 (or higher) support in GPU driver or CPU runtime

Linux

For Ubuntu >= 16 you can use:

- `sudo add-apt-repository ppa:robbert-harms/cbclab`
- `sudo apt-get update`
- `sudo apt-get install python3-mct`

For Debian users and Ubuntu < 16 users, install MDT with:

- `sudo apt-get install python3 python3-pip python3-pyopencl python3-numpy python3-nibabel python3-pyqt5 python3-matplotlib python3-yaml python3-argcomplete libpng-dev libfreetype6-dev libxft-dev`
- `sudo pip3 install mct`

Note that `python3-nibabel` may need NeuroDebian to be available on your machine. An alternative is to use `pip3 install nibabel` instead.

Windows

The installation on Windows is a little bit more complex and the following is only a quick reference guide. To save duplication of information and since this package depends on MDT and MOT, the complete install instructions can be copied from the MDT documentation. After following that guide, installation of MCT is simply done using `pip install mct`. The quick overview is:

- Install Anaconda Python 3.5

- Install MOT using the guide at https://mot.readthedocs.io
- Open an Anaconda shell and type: `pip install mct`

Mac

- Install Anaconda Python 3.5
- Open a terminal and type: `pip install mct`

Please note that Mac support is experimental due to the unstable nature of the OpenCL drivers in Mac, that is, users running MDT with the GPU as selected device may experience crashes. Running MDT in the CPU seems to work though.

For more information and full installation instructions please see the documentation of the MDT package https://maastrichtdiffusiontoolbox.readthedocs.org

1.4 Roadmap

1) Add a few more reconstruction methods like:
   - Roemer
   - GRAPPA
   - SENSE
2) Improve the data handling and memory usage.

Changelog

Changelog

v0.2.11 (2019-02-19)

- Following changes in MDT.

v0.2.10 (2018-10-24)

- Rebased against the latest MOT and MDT versions.

v0.2.9 (2018-08-23)

- Following changes in MOT.

v0.2.8 (2018-08-17)

- Removed redundant super arguments.
- Following changes in MOT and MDT.

v0.2.7 (2018-08-02)

- Regression fix.

v0.2.6 (2018-08-02)

- Following changes in MOT.
- Removed six as a dependency.

v0.2.5 (2018-07-17)

- Updated makefile to use twine for uploading to PyPi.

v0.2.4 (2018-05-03)

- Following changes in MOT.

v0.2.3 (2018-04-11)

- Following changes in MOT.

v0.2.2 (2018-04-09)

- Following changes in MOT and MDT.

v0.2.1 (2018-04-04)

Added

- Adds post-optimization transformation to STARC.

Changed

- Updates following the changes in MOT.

Other

- Merge branch ‘master’ of github.com:cbclab/MCT.
- Version bump.
- Following changes in MOT.

v0.2 (2017-09-22)

Added

- Adds rCovSoS, improvements to rSoS. Adds calculation method for calculating the noise covariance matrix.
- Adds the rCovSoS method.
- Adds a split volume script to split single volumes into channels.

Changed

- Updates to the comments.
- Changed the whole restructiring pipelines to make room for GRAPPA and SENSE.

Other

- Prepared next release. Added a volumes switch to the reconstruction method.
- Small refactoring in the API, made the channels a constructor parameter.
- Merge branch 'master' of https://github.com/cbclab/MCT.
- Reverted the data loading again. This version uses more memory.
- Processing version with different kind of memory loading.
- Removed the multiprocessing in favor of single threaded data loading. It is more robust.
- Project rename.

v0.1.1 (2017-09-12)

Added

- Adds cl-device index flag and max batch size flag to the mct-reconstruct CLI.

Other

- Prepared next release.

v0.1.0 (2017-09-12)

Added

- Adds changelog.

Changed

- Updates to docs.
- Updates to docs.
- Updates to docs.
- Updates to docs.
- Updates to readme.

Other

- Small update to docs.
- Small update to docs.
- Prepared for first release.

- First public release.

Developers

The MRI Coil-combine Toolbox is a coil-combine toolbox meant for MRI reconstruction. Software development by Robbert Harms, under the (Phd) supervision of Alard Roebroeck, at Maastricht University.

List of contributors:

- **Robbert Harms**
  - Lead developer
- **Alard Roebroeck**
  - Phd. supervision
- **Francisco Fritz**
  - Quality control
- **Sriranga Kashyap**
  - Work on STARC, quality control