# **TIMBIR Documentation**

Release 0.0.2

**Purdue University** 

April 15, 2016

Contents
----------

Install directions	3
Data Collection	5
Examples	7
Development	9
Frequently asked questions	15
Credits	17
Indices and tables	19
bliography	21
	Data CollectionExamplesDevelopmentFrequently asked questionsCreditsIndices and tables

The Time-Interlaced Model-Based Iterative Reconstruction (TIMBIR) is a method for 4D time-space reconstruction of data acquired using synchrotron X-ray computed tomography.

This guide is maintained on GitHub.

The Time-Interlaced Model-Based Iterative Reconstruction (TIMBIR) [[A3], [A2]] is a method for 4D time-space reconstruction of data acquired using synchrotron X-ray computed tomography. TIMBIR is a synergistic combination of two innovations. The first innovation, interlaced view sampling, is a novel method of data acquisition which distributes the view angles more evenly in time. The second innovation is a 4D model-based iterative reconstruction algorithm (MBIR) which can produce time-resolved volumetric reconstruction of the sample from the interlaced views. In addition to modeling both the sensor noise statistics and the 4D object, the MBIR algorithm also reduces ring and streak artifacts by more accurately modeling the measurement non-idealities [(A3), (A2), (A1)].

#### Install directions

This section covers the basics of how to download and install TIMBIR.

#### **Contents:**

- Dependencies
- Dependencies Install
- Compiling TIMBIR
- Running the reconstruction algorithm

#### **1.1 Dependencies**

- MPI compiler (Intel or Open MPI)
- OpenMP
- make utility

# **1.2 Dependencies Install**

• Install MPI (Here I use openMPI as a example):

```
$ wget https://www.open-mpi.org/software/ompi/v1.10/downloads/openmpi-1.10.2.tar.gz
$ tar xvzf openmpi-1.10.2.tar.gz
$ cd <openmpi path>
$ ./configure --prefix=<your mpi install path>
$ make all install
```

• Install HDF5 library. First go to https://www.hdfgroup.org/HDF5/release/obtainsrc.html#conf to find the appropriate hdf5 library for your platform (here we use hdf5-1.8.16.tar as example:

```
$ tar xvf hdf5-1.8.16.tar
$ cd <hdf5 path>
$ ./configure --prefix=/clhome/KYUE/lib/hdf5 --enable-fortran --enable-cxx
$ make
$ make
$ make install
```

• Set your library path with HDF5 library and MPI library (here we use bash as example):

```
$ vi env.sh (create a bash script)
$ export HDF5_BASE=<hdf5 full path>
$ export MPI_BASE=<MPI full path>
$ export PATH = ${MPI_BASE}/bin:${HDF5_BASE}/bin:$PATH
$ export LD_LIBRARY_PATH= = ${MPI_BASE}/lib:${HDF5_BASE}/lib64:$PATH
$ source env.sh
```

# 1.3 Compiling TIMBIR

To compile the MBIR algorithm code:

```
$ git clone https://github.com/adityamnk/timbir.git timbir
$ cd timbir/src/MBIR_4D
```

This generates library files in timbir/src/lib. For more information, read the README in timbir/src/MBIR\_4D.

## 1.4 Running the reconstruction algorithm

If the input data format is a standard binary, run the code in timbir/src/reconstruct/bin\_data. For more information on data format and running the code, read the README in timbir/src/reconstruct/basic:

```
$ cd timbir/src/reconstruct/bin_data
$ make
```

This generates executables in the same folder.

If the input data is in HDF format used at APS, run the code in timbir/src/reconstruct/aps\_data. For more information on data format and running the code, read the README in timbir/src/reconstruct/aps\_data:

```
$ cd timbir/src/reconstruct/aps_data
$ make
```

This generates executables in the same folder.

If the input data is in standard HDF format, run the code in timbir/src/reconstruct/std\_data. For more informantion on data format and running the code, read the README in timbir/src/reconstruct/std\_data:

```
$ cd timbir/src/reconstruct/std_data
$ make #Generates executables in the same folder
```

# **Data Collection**

This section covers the basics of how to collect the data that will be analysed by TIMBIR including data collection scripts run at the APS.

#### **Contents:**

• to be completed

# 2.1 to be completed

### **Examples**

In this section, we list scripts that can be used to compile the code and run the reconstruction algorithm. Examples are provided for both 3D and 4D reconstruction on either a unix based system (Linux/Mac) or a super-computing cluster.

### 3.1 3D Reconstruction of Shepp-Logan Phantom

- Unix/Linux/Mac OS:: Run the script timbir/demo/recon\_3d/shepp-logan-3D/run\_unix.sh
- Super-computing cluster (Rice cluster at Purdue):: Run the script timbir/demo/recon\_3d/shepp-logan-3D/run\_cluster.sh

### 3.2 4D Reconstruction of Cahn-Hilliard Phantom

- Unix/Linux/Mac OS:: Run the script timbir/demo/recon\_4d/cahn-hilliard-4D/run\_unix.sh
- Super-computing cluster (Rice cluster at Purdue):: Run the script timbir/demo/recon\_4d/cahn-hilliard-4D/run\_cluster.sh

### Development

This section explains the basics for developers who wish to contribute to the TIMBIR project.

#### **Contents:**

- *Cloning the repository*
- Coding conventions
- Package versioning
- Commiting changes
- Contributing back

# 4.1 Cloning the repository

The project is maintained on GitHub, which is a version control and a collaboration platform for software developers. To start first register on GitHub and fork the TIMBIR repository by clicking the **Fork** button in the header of the TIMBIR repository:

This repository Sear	ch	Explore Gist Blog H	lelp 🛓	dgursoy 🕂 🖬 🛱 🕞
dgursoy / tome forked from tomopy/tomop			Unwatch + 1	★ Star 0 <b>% Fork</b>
maging toolbox. — Ed	it			<> Code
🕞 527 commits	4 branches	⊗ 3 releases	15 contributors	
				17 Pull requests
P branch: master -	tomopy / +		i≡	II Wiki
This branch is 45 commits ah	ead of tomopy:master		🕅 Pull Request  한 Compare	
MNT: doc corrections				Pulse
dgursoy authored 13 ho	urs ago		latest commit 8de1d830c7 🔂	III Graphs
doc	MNT: doc corrections		13 hours ago	
tomopy	MNT: doc corrections		13 hours ago	X Settings
.gitignore	MNT: bib test		a day ago	
.project	refs #11: make project easier to	edit with eclipse	a year ago	https://github.com/(
.pydevproject	refs #11: make project easier to	edit with eclipse	a year ago	You can clone with HTTPS, SSH,
LICENSE.txt	ENH: code improvements		7 days ago	or Subversion. ()
MANIFEST.in	ENH: typo corrected		7 days ago	Clone in Desktop
README.rst	ENH: typo corrected		7 days ago	C Download ZIP
	ENH: code improvements		7 days ago	
🗎 bld.bat	ENH: code improvements		7 days ago	
build.sh	ENH: code improvements		7 days ago	

This successfully creates a copy of the project in your personal GitHub space. The next thing you want to do is to clone it to your local machine. You can do this by clicking the **Clone in Desktop** button in the bottom of the right hand side bar:

This repository Sea	rch	Explore Gist Blog H	elp 🔬 d	gursoy 🕂 🖬 🔂 🕞
dgursoy / tom forked from tomopy/tomop			Unwatch - 1	★ Star 0 ¥ Fork 25
maging toolbox. — Ec	dit			<> Code
🕞 527 commits		⊗ 3 releases	😚 15 contributors	<> Code
				11 Pull requests
🗘 🕻 branch: master	tomopy / +		:=	III Wiki
This branch is 45 commits al	head of tomopy:master		1) Pull Request 🗄 Compare	
MNT: doc corrections				-/~ Pulse
dgursoy authored 13 h	ours ago		latest commit 8de1d830c7 🔂	II Graphs
doc	MNT: doc corrections		13 hours ago	
tomopy	MNT: doc corrections		13 hours ago	3 Settings
.gitignore	MNT: bib test		a day ago	
.project	refs #11: make project easier to	edit with eclipse	a year ago	HTTPS clone URL
.pydevproject	refs #11: make project easier to	edit with eclipse	a year ago	You can clone with HTTPS, SSH,
LICENSE.txt	ENH: code improvements		7 days a	Clone in Desktop
MANIFEST.in	ENH: typo corrected		7 days ag	E Cione in Desktop
README.rst	ENH: typo corrected		7 days ago	C Download ZIP
	ENH: code improvements		7 days ago	
D	ENH: code improvements		7 days ago	
bld.bat				

This will launch the GitHub desktop application (available for both Mac and Win) and ask you where you want to save it. Select a location in your computer and feel comfortable with making modifications in the code.

## 4.2 Coding conventions

We try to keep a consistent and readable code. So, please keep in mind the following style and syntax guidance before you start coding.

First of all the code should be well documented, easy to understand, and integrate well into the rest of the project. For example, when you are writing a new function always describe the purpose and the parameters:

```
def my_awesome_func(a, b):
    """
    Adds two numbers.
    Parameters
    ------
    a : scalar (float)
        First number to add
    b : scalar (float)
        Second number to add
    Returns
```

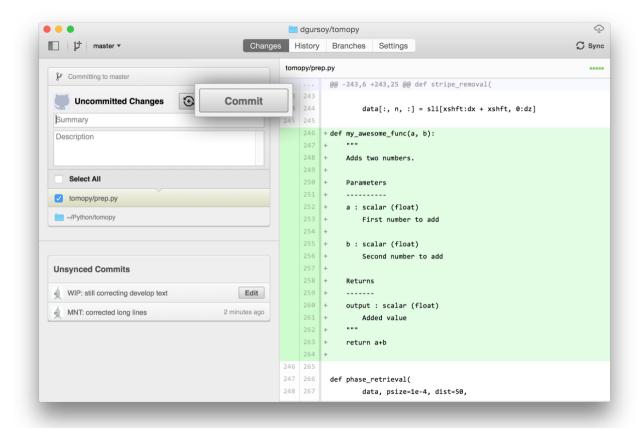
```
output : scalar (float)
Added value
"""
return a+b
```

## 4.3 Package versioning

We follow the X.Y.Z (Major.Minor.Patch) semantic for package versioning. The version should be updated before each pull request accordingly. The patch number is incremented for minor changes and bug fixes which do not change the software's API. The minor version is incremented for releases which add new, but backward-compatible, API features, and the major version is incremented for API changes which are not backward-compatible. For example, software which relies on version 2.1.5 of an API is compatible with version 2.2.3, but not necessarily with 3.2.4.

### 4.4 Commiting changes

After making some changes in the code, you may want to take a *snapshot* of the edits you made. That's when you make a *commit*. To do this, launch the GitHub desktop application and it should provide you all the changes in your code since your last commit. Write a brief *Summary* and *Description* about the changes you made and click the **Commit** button:



You can continue to make changes, add modules, write your own functions, and take more *Commit snapshots* of your code writing process.

# 4.5 Contributing back

Once you feel that the functionality you added would benefit the community, then you should consider contributing back to the TIMBIR project. For this, go to your online GitHub repository of the project and click on the *green* button to compare, review and create a pull request.

This repository Se		Explore Gist Blog H		gursoy 🕂 🖬 🖬 🕻
dgursoy / ton forked from tomopy/tome			③ Unwatch ▼ 1	★ Star 0 % Fork 2
Imaging toolbox. — E	dit			
3527 commits		⊗ 3 releases	15 contributors	<> Code
<b>b</b>				1) Pull requests
P branch: master	tomopy / +		E	III Wiki
This branch is 45 commits	ahead of tomopy:master		🖺 Pull Request  🗄 Compare	
MNT: doc corrections				-/~ Pulse
dgursoy authored 13	nours ago		latest commit 8de1d830c7 🔂	III Graphs
doc	MNT: doc corrections		13 hours ago	
tomopy	MNT: doc corrections		13 hours ago	X Settings
.gitignore	MNT: bib test		a day ago	
.project	refs #11: make project easier to	edit with eclipse	a year ago	HTTPS clone URL
.pydevproject	refs #11: make project easier to	edit with eclipse	a year ago	You can clone with HTTPS, SSH
LICENSE.txt	ENH: code improvements		7 days ago	or Subversion. ③
MANIFEST.in	ENH: typo corrected		7 days ago	Clone in Desktop
README.rst	ENH: typo corrected		7 days ago	$\Rightarrow$ Download ZIP
	ENH: code improvements		7 days ago	
bld.bat	ENH: code improvements		7 days ago	
			7 days ago	

After clicking on this button, you are presented with a review page where you can get a high-level overview of what exactly has changed between your forked branch and the original TIMBIR repository. When you're ready to submit your pull request, click **Create pull request**:

	mopy / <b>tomo</b>	ру		O Unwatch	n → 31 🛨 Star	14 V Fork
	nparing of two branches to s	changes ee what's changed or to start a new	pull request. If you need to, you	u can also compare acr	oss forks.	<
រោ	base fork: tomopy	/tomopy - base: master	head fork: dgursoy/tomopy -	compare: master -		
~	Able to merge. T	hese branches can be automatically me	erged.			8
Create n	oull request					
energie p	Juli request	Discuss and review the changes	in this comparison with others.			· ·
	_		in this comparison with others.		ŝ	1 contributor
-> Con	_	Files changed 192 PCommit			¢	1
-> Con	mmits 45 🗈	Files changed 192 PCommit	comments 0		¢	1
← Con	mmits 45 🗈	Files changed 192 Files changed 192	comments 0		¢	3 1 contributor
Con	mmits 45 🕑 mmits on Mar 29, 2 dgursoy dgursoy	Files changed 192 Commit 015 ENH: code improveme	comments o		¢	7b8ec13
-> Con ↓ Con → ↓	mmits 45 🕑 mmits on Mar 29, 2 dgursoy dgursoy dgursoy	Files changed 192 Commit 015 ENH: code improveme ENH: typo corrected	comments o		ŝ	7b8ec13 ecf058e
<ul> <li>◆ Con</li> <li>↓</li> <li>↓<!--</td--><td>mmits 45 💽 mmits on Mar 29, 2 dgursoy dgursoy dgursoy dgursoy dgursoy</td><td>Files changed 192 Commit 015 ENH: code improveme ENH: typo corrected ENH: typo corrected</td><td>comments o</td><td></td><td>ĝ</td><td>7b8ec13 ecf058e 79e6382</td></li></ul>	mmits 45 💽 mmits on Mar 29, 2 dgursoy dgursoy dgursoy dgursoy dgursoy	Files changed 192 Commit 015 ENH: code improveme ENH: typo corrected ENH: typo corrected	comments o		ĝ	7b8ec13 ecf058e 79e6382
<ul> <li>Con</li> <li>Ip</li> <li>Con</li> <l< td=""><td>mmits 45 💽 mmits on Mar 29, 2 dgursoy dgursoy dgursoy dgursoy dgursoy</td><td>Files changed 192 Commit 015 ENH: code improveme ENH: typo corrected ENH: typo corrected ENH: typo corrected</td><td>comments o</td><td>•</td><td>đ</td><td>7b8ec13 ecf858e 79e6382 f102d5e</td></l<></ul>	mmits 45 💽 mmits on Mar 29, 2 dgursoy dgursoy dgursoy dgursoy dgursoy	Files changed 192 Commit 015 ENH: code improveme ENH: typo corrected ENH: typo corrected ENH: typo corrected	comments o	•	đ	7b8ec13 ecf858e 79e6382 f102d5e
<ul> <li>◆ Con</li> <li>⑤ Con</li> <li>◆ ◆</li> <li>◆ ◆<!--</td--><td>mmits 45 💽 mmits on Mar 29, 2 dgursoy dgursoy dgursoy dgursoy dgursoy dgursoy dgursoy</td><td>Files changed 192 Commit 015 ENH: code improveme ENH: typo corrected ENH: typo corrected ENH: typo corrected ENH: typo corrected ENH: typo corrected</td><td>comments o</td><td>•</td><td>đ</td><td>7b8ec13 ecf658e 79e6382 f102d5e alfaae0</td></li></ul>	mmits 45 💽 mmits on Mar 29, 2 dgursoy dgursoy dgursoy dgursoy dgursoy dgursoy dgursoy	Files changed 192 Commit 015 ENH: code improveme ENH: typo corrected ENH: typo corrected ENH: typo corrected ENH: typo corrected ENH: typo corrected	comments o	•	đ	7b8ec13 ecf658e 79e6382 f102d5e alfaae0
◆ Con	mmits 45 💽 mmits on Mar 29, 2 dgursoy dgursoy dgursoy dgursoy dgursoy dgursoy dgursoy dgursoy	Files changed 192 Commit 015 ENH: code improveme ENH: typo corrected ENH: typo corrected ENH: typo corrected ENH: typo corrected ENH: typo corrected ENH: typo corrected	comments o	•	÷	7b8ec13 ecf858e 79e6382 f102d5e a1faæ0 22b0e92

Clicking on **Create pull request** sends you to a discussion page, where you can enter a title and optional description. It's important to provide as much useful information and a rationale for why you're making this Pull Request in the first place.

When you're ready typing out your heartfelt argument, click on Send pull request. You're done!

# Frequently asked questions

Here's a list of questions.

Questions

• How can I report bugs?

# 5.1 How can I report bugs?

The easiest way to report bugs or get help is to open an issue on GitHub. Simply go to the project GitHub page, click on Issues in the right menu tab and submit your report or question.

# Credits

We kindly request that you cite the following articles if you use TIMBIR:

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

#### Bibliography

- [A1] K.A. Mohan, S.V. Venkatakrishnan, L.F. Drummy, J. Simmons, D.Y. Parkinson, and C.A. Bouman. Model-based iterative reconstruction for synchrotron x-ray tomography. In *Acoustics, Speech and Signal Processing (ICASSP)*, 2014 IEEE International Conference on, 6909–6913. May 2014. doi:10.1109/ICASSP.2014.6854939.
- [A2] K.A. Mohan, S.V. Venkatakrishnan, J.W. Gibbs, E.B. Gulsoy, Xianghui Xiao, M. De Graef, P.W. Voorhees, and C.A. Bouman. 4d model-based iterative reconstruction from interlaced views. In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, 783–787. April 2015. doi:10.1109/ICASSP.2015.7178076.
- [A3] Kadri Mohan, Singanallur Venkatakrishnan, John Gibbs, E Gulsoy, Xianghui Xiao, Marc De Graef, Peter Voorhees, and Charles Bouman. Timbir: a method for time-space reconstruction from interlaced views. *IEEE Transactions on Computational Imaging*, 2015.