

NeXtlook

Darryn Jordan

June 5, 2017

# Contents

<b>1</b>	<b>Installation</b>	<b>1</b>
1.1	Prerequisites . . . . .	1
1.1.1	FFTw . . . . .	1
1.1.2	Misc. . . . .	1
1.2	NeXtlook . . . . .	1
<b>2</b>	<b>Usage</b>	<b>3</b>
2.1	Experiment Configuration . . . . .	3
2.1.1	Example Parameter File . . . . .	3
2.1.2	Parameter Description . . . . .	4

## **Abstract**

NeXtlook is a quicklook processor for generating range-time-intensity and range-Doppler plots of NeXtRAD datasets. Multithreading is supported for pulse-compression, but Doppler-processing is currently only supported when using a single thread.

# Chapter 1

## Installation

*This chapter documents the installation process for the NeXtlook processor.*

### 1.1 Prerequisites

#### 1.1.1 FFTw

Download the latest version of FFTw from their website ensuring that the version number is 3.3.6 or higher. This version introduced thread-safe planners which are vital. Extract the files in an arbitrary location, enter the extracted folder and run the following commands,

```
./configure --enable-threads  
make -j  
sudo make install
```

If any problems occur I suggest using the following command to perform a clean before retrying,

```
make distclean
```

The extracted folder can now be deleted.

#### 1.1.2 Misc.

```
sudo apt install cmake  
sudo apt install git  
sudo apt install libopencv-dev  
sudo apt install libboost-all-dev
```

### 1.2 NeXtlook

At the time of writing, the NeXtlook source code is located at <https://github.com/darrynjordan/nextlook>. The source can be downloaded as a zip folder or cloned from the repo,

```
git clone https://github.com/darrynjordan/nextlook.git
```

Within the cloned folder, create a new folder to hold the built executable and run Cmake,

```
mkdir build  
cd build  
cmake ..
```

If no errors have occurred, the executable can be compiled and run as follows,

```
make -j  
./nextlook
```

# Chapter 2

## Usage

### 2.1 Experiment Configuration

On first run, the program will probably complain about not finding the binary dataset. Do not panic. Within the root folder there should be a textfile called **experiment.ini** which contains all experiment parameters. This is where the path to the binary dataset and reference waveform, and other options must be configured.

#### 2.1.1 Example Parameter File

```
[config]
debug_mode = false

[dataset]
data_filename = ../12_12_2016_14_48_59_adc1data.dat
n_cmplx_samples_range_line = 2048
n_range_lines = 30000
ref_filename = ../Ref_PL_Xband_MPALNA_3us.dat
n_cmplx_samples_ref = 297
n_cmplx_samples_padded = 2048;

[processing]
n_threads = 1;
doppler_enabled = true
doppler_cpi = 256
range_window = UNIFORM
doppler_window = UNIFORM

[visualisation]
update_rate = 1024
```

```
doppler_colour_map = 1
rti_colour_map = 2
histogram_equalization = 1
slow = 0
threshold = 50
```

### 2.1.2 Parameter Description

#### config

- **debug-mode** provides additional information through the command line during processing.

#### dataset

- **data-filename** the path to the binary dataset to be processed.
- **n-cmplx-samples-range-line** number of complex samples in each range line.
- **n-range-lines** total number of range lines to process.
- **ref-filename** the path to the binary reference to be processed.
- **n-cmplx-samples-ref** the number of complex samples in the reference waveform.
- **n-cmplx-samples-padded** determines the number of complex samples each range line must be zero padded to.

#### processing

- **n-threads** the number of threads which will be launched for pulse compression.
- **doppler-enabled** toggles Doppler processing.
- **doppler-cpi** number of range bins used for each Doppler plot.
- **range-window** tapering function applied to each range line.
- **doppler-window** tapering function applied to each Doppler line.

#### visualisation

- **update-rate** number of range lines to be processed before a plot update occurs.
- **doppler-colour-map** [0 - 11] index of colour map to be applied to range-Doppler plot.

- **rti-colour-map** [0 - 11] index of colour map to be applied to RTI plot.
- **histogram-equalization** [0 - 1] toggles histogram equalisation for both plots.
- **slow** [0 - 500] number of milliseconds processing delay between each plot update.
- **threshold** [0 - 255] apply threshold to both plots.