

## GPU Multirate FIR Filter

Generated by Doxygen 1.8.6

Fri Sep 16 2016 15:10:22



# Contents

<b>1</b>	<b>Data Structure Index</b>	<b>1</b>
1.1	Data Structures . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Structure Documentation</b>	<b>5</b>
3.1	cmd_args Struct Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Field Documentation . . . . .	5
3.1.2.1	insize . . . . .	5
3.1.2.2	nf . . . . .	5
3.1.2.3	nrates . . . . .	6
3.1.2.4	rconst . . . . .	6
3.1.2.5	tim . . . . .	6
3.2	filter_arrays Struct Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.2.2	Field Documentation . . . . .	6
3.2.2.1	bk . . . . .	6
3.2.2.2	buf_in . . . . .	6
3.2.2.3	buf_out . . . . .	7
3.2.2.4	m_offb . . . . .	7
3.3	gpu_arrays Struct Reference . . . . .	7
3.3.1	Detailed Description . . . . .	7
3.3.2	Field Documentation . . . . .	7
3.3.2.1	d_filters . . . . .	7
3.3.2.2	d_in . . . . .	8
3.3.2.3	d_out . . . . .	8
3.3.2.4	h_in . . . . .	8
3.3.2.5	h_out . . . . .	8
3.3.2.6	isize . . . . .	8
3.3.2.7	osize . . . . .	8

3.3.2.8	stream	8
3.4	params Struct Reference	8
3.4.1	Detailed Description	9
3.4.2	Field Documentation	9
3.4.2.1	fsize	9
3.4.2.2	nfilters	9
3.4.2.3	nrates	9
3.4.2.4	rnumf	9
3.4.2.5	streams	9
<b>4</b>	<b>File Documentation</b>	<b>11</b>
4.1	filtermain.cpp File Reference	11
4.1.1	Function Documentation	11
4.1.1.1	main	11
4.2	filters.h File Reference	11
4.2.1	Macro Definition Documentation	12
4.2.1.1	A_SIZE	12
4.2.1.2	B_SIZE	12
4.2.1.3	CUDA_SHM	12
4.2.1.4	MAX_FILTERS	12
4.2.1.5	MAX_RATES	13
4.2.1.6	OFFSET	13
4.2.1.7	SIZE_MUL	13
4.2.2	Function Documentation	13
4.2.2.1	check_results	13
4.2.2.2	compute_omp	13
4.2.2.3	compute_ref	13
4.2.2.4	print_usage	14
4.2.2.5	read_command_line	14
4.2.3	Variable Documentation	14
4.2.3.1	rdiv	14
4.3	filters_cuda.cu File Reference	14
4.3.1	Function Documentation	14
4.3.1.1	cudaMultiFilterFirClose	14
4.3.1.2	cudaMultiFilterFirInit	15
4.3.1.3	cudaMultiFilterFirStreams	15
4.3.1.4	multiFilterGpuDevice_2dgrid	15
4.3.1.5	multiFilterGpuDevice_2dgrid_shmem	15
4.4	filters_host.cpp File Reference	15
4.4.1	Function Documentation	16

---

4.4.1.1	<a href="#">check_results</a>	16
4.4.1.2	<a href="#">compute_omp</a>	16
4.4.1.3	<a href="#">compute_ref</a>	17
4.4.1.4	<a href="#">filterCpuFir</a>	18
4.4.1.5	<a href="#">print_usage</a>	18
4.4.1.6	<a href="#">read_command_line</a>	18
<b>Index</b>		<b>19</b>



# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">cmd_args</a>	5
<a href="#">filter_arrays</a>	6
<a href="#">gpu_arrays</a>	7
<a href="#">params</a>	8





## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">filtermain.cpp</a>	11
<a href="#">filters.h</a>	11
<a href="#">filters_cuda.cu</a>	14
<a href="#">filters_host.cpp</a>	15



## Chapter 3

# Data Structure Documentation

### 3.1 cmd\_args Struct Reference

```
#include <filters.h>
```

#### Data Fields

- int `nrates`  
*how many sampling rates to process*
- int `nf`  
*total number of filters to process*
- int `insize`  
*input size before resampling*
- int `rconst`  
*for nrates=1 keep the initial input size*
- int `tim`  
*time process*

#### 3.1.1 Detailed Description

Definition at line 67 of file filters.h.

#### 3.1.2 Field Documentation

##### 3.1.2.1 int cmd\_args::insize

input size before resampling

Definition at line 70 of file filters.h.

##### 3.1.2.2 int cmd\_args::nf

total number of filters to process

Definition at line 69 of file filters.h.

### 3.1.2.3 int cmd\_args::nrates

how many sampling rates to process

Definition at line 68 of file filters.h.

### 3.1.2.4 int cmd\_args::rconst

for nrates=1 keep the initial input size

Definition at line 71 of file filters.h.

### 3.1.2.5 int cmd\_args::tim

time process

Definition at line 72 of file filters.h.

The documentation for this struct was generated from the following file:

- [filters.h](#)

## 3.2 filter\_arrays Struct Reference

```
#include <filters.h>
```

### Data Fields

- float [buf\\_in](#) [[MAX\\_FILTERS](#)][[B\\_SIZE+OFFSET](#)]  
*CPU buffer.*
- float [buf\\_out](#) [[MAX\\_FILTERS](#)][[A\\_SIZE+OFFSET](#)]  
*for IIR filter*
- int [m\\_offb](#) [[MAX\\_FILTERS](#)]  
*offset counter for CPU buffers*
- float \* [bk](#)  
*filter coefficients array*

### 3.2.1 Detailed Description

Definition at line 48 of file filters.h.

### 3.2.2 Field Documentation

#### 3.2.2.1 float\* filter\_arrays::bk

filter coefficients array

Definition at line 52 of file filters.h.

#### 3.2.2.2 float filter\_arrays::buf\_in[MAX\_FILTERS][B\_SIZE+OFFSET]

CPU buffer.

Definition at line 49 of file filters.h.

## 3.2.2.3 float filter\_arrays::buf\_out[MAX\_FILTERS][A\_SIZE+OFFSET]

for IIR filter

Definition at line 50 of file filters.h.

## 3.2.2.4 int filter\_arrays::m\_offb[MAX\_FILTERS]

offset counter for CPU buffers

Definition at line 51 of file filters.h.

The documentation for this struct was generated from the following file:

- [filters.h](#)

### 3.3 gpu\_arrays Struct Reference

```
#include <filters.h>
```

#### Data Fields

- float \* [d\\_filters](#) [[MAX\\_RATES](#)]  
*filters to be stored on the GPU*
- float \* [d\\_in](#) [[MAX\\_RATES](#)]  
*GPU input array for each sampling rate.*
- int [isize](#) [[MAX\\_RATES](#)]  
*input size for each sampling rate*
- float \* [d\\_out](#) [[MAX\\_RATES](#)]  
*GPU output array for each sampling rate.*
- int [osize](#) [[MAX\\_RATES](#)]  
*total output size for each set of filters*
- float \* [h\\_in](#) [[MAX\\_RATES](#)]  
*host input arrays for GPU transfer*
- float \* [h\\_out](#) [[MAX\\_RATES](#)]  
*host output arrays for GPU transfer*
- cudaStream\_t [stream](#) [[MAX\\_RATES](#)]  
*CUDA stream objects.*

#### 3.3.1 Detailed Description

Definition at line 55 of file filters.h.

#### 3.3.2 Field Documentation

## 3.3.2.1 float\* gpu\_arrays::d\_filters[MAX\_RATES]

filters to be stored on the GPU

Definition at line 56 of file filters.h.

### 3.3.2.2 `float* gpu_arrays::d_in[MAX_RATES]`

GPU input array for each sampling rate.

Definition at line 57 of file filters.h.

### 3.3.2.3 `float* gpu_arrays::d_out[MAX_RATES]`

GPU output array for each sampling rate.

Definition at line 59 of file filters.h.

### 3.3.2.4 `float* gpu_arrays::h_in[MAX_RATES]`

host input arrays for GPU transfer

Definition at line 61 of file filters.h.

### 3.3.2.5 `float* gpu_arrays::h_out[MAX_RATES]`

host output arrays for GPU transfer

Definition at line 62 of file filters.h.

### 3.3.2.6 `int gpu_arrays::isize[MAX_RATES]`

input size for each sampling rate

Definition at line 58 of file filters.h.

### 3.3.2.7 `int gpu_arrays::osize[MAX_RATES]`

total output size for each set of filters

Definition at line 60 of file filters.h.

### 3.3.2.8 `cudaStream_t gpu_arrays::stream[MAX_RATES]`

CUDA stream objects.

Definition at line 63 of file filters.h.

The documentation for this struct was generated from the following file:

- [filters.h](#)

## 3.4 params Struct Reference

```
#include <filters.h>
```

### Data Fields

- `int nfilters`  
*total number of filters*
- `int fsize`

- filter size*
  - int [nrates](#)  
*number of input sampling rates to process*
  - int [rnumf](#) [[MAX\\_RATES](#)]  
*number of filters for each sampling rate*
  - int [streams](#)  
*use streams?*

### 3.4.1 Detailed Description

Definition at line 38 of file filters.h.

### 3.4.2 Field Documentation

#### 3.4.2.1 int params::fsize

filter size

Definition at line 40 of file filters.h.

#### 3.4.2.2 int params::nfilters

total number of filters

Definition at line 39 of file filters.h.

#### 3.4.2.3 int params::nrates

number of input sampling rates to process

Definition at line 41 of file filters.h.

#### 3.4.2.4 int params::rnumf[[MAX\\_RATES](#)]

number of filters for each sampling rate

Definition at line 42 of file filters.h.

#### 3.4.2.5 int params::streams

use streams?

Definition at line 43 of file filters.h.

The documentation for this struct was generated from the following file:

- [filters.h](#)





## Chapter 4

# File Documentation

### 4.1 filtermain.cpp File Reference

```
#include <stdio>
#include <stdlib>
#include <string>
#include <sys/time.h>
#include "filters.h"
```

#### Functions

- int [main](#) (int argc, char \*\*argv)

#### 4.1.1 Function Documentation

##### 4.1.1.1 int main ( int *argc*, char \*\* *argv* )

Definition at line 29 of file filtermain.cpp.

### 4.2 filters.h File Reference

```
#include <cuda_runtime.h>
```

#### Data Structures

- struct [params](#)
- struct [filter\\_arrays](#)
- struct [gpu\\_arrays](#)
- struct [cmd\\_args](#)

#### Macros

- #define [CUDA\\_SHM](#) 1  
*Use shared memory GPU kernel.*
- #define [SIZE\\_MUL](#) 1

- Multiplier of 11 to derive filter length.*

  - `#define B_SIZE 11 * SIZE_MUL`

*filter length*
- `#define A_SIZE (B_SIZE - 1)`

*For IIR filter. not used currently.*
- `#define OFFSET 20`

*Offset for CPU filter input buffer.*
- `#define MAX_RATES 10`

*Maximum number of sampling rates.*
- `#define MAX_FILTERS 1024`

*Maximum number of filters.*

## Functions

- void `compute_ref` (float \*h\_in[], float \*h\_reference[], `gpu_arrays` \*gpuarrays, `params` \*gparams, `cmd_args` \*args, `filter_arrays` \*farr, int N)
- void `compute_omp` (float \*h\_in[], float \*h\_reference[], `gpu_arrays` \*gpuarrays, `params` \*gparams, `cmd_args` \*args, `filter_arrays` \*farr, int N)
- void `check_results` (float \*h\_reference[], float \*h\_out[], `gpu_arrays` \*gpuarrays, `params` \*gparams, int N)
- void `read_command_line` (int argc, char \*argv[], `cmd_args` \*args)
- void `print_usage` ()

## Variables

- static int `rdiv` [`MAX_RATES`] = {2, 10, 50, 4, 16, 32, 8, 24, 36, 42}
- Decimation factors for multiple input rates.*

### 4.2.1 Macro Definition Documentation

#### 4.2.1.1 `#define A_SIZE (B_SIZE - 1)`

For IIR filter. not used currently.

Definition at line 30 of file filters.h.

#### 4.2.1.2 `#define B_SIZE 11 * SIZE_MUL`

filter length

Definition at line 28 of file filters.h.

#### 4.2.1.3 `#define CUDA_SHM 1`

Use shared memory GPU kernel.

Definition at line 25 of file filters.h.

#### 4.2.1.4 `#define MAX_FILTERS 1024`

Maximum number of filters.

Definition at line 35 of file filters.h.

## 4.2.1.5 #define MAX\_RATES 10

Maximum number of sampling rates.

Definition at line 34 of file filters.h.

## 4.2.1.6 #define OFFSET 20

Offset for CPU filter input buffer.

Definition at line 32 of file filters.h.

## 4.2.1.7 #define SIZE\_MUL 1

Multiplier of 11 to derive filter length.

Definition at line 27 of file filters.h.

## 4.2.2 Function Documentation

4.2.2.1 void check\_results ( float \* *h\_reference*[], float \* *h\_out*[], gpu\_arrays \* *gpuarrays*, params \* *gparams*, int *N* )

Check the relative error between CPU and GPU filter computation results

Parameters

<i>h_in</i>	array containing all input blocks of the MR filter
<i>h_reference</i>	array containing the MR filter CPU output
<i>h_out</i>	array containing the MR filter GPU output
<i>gpuarrays</i>	structure containing the gpu arrays and parameters
<i>gparams</i>	structure containing parameters
<i>N</i>	the number of input blocks to be processed.

Definition at line 188 of file filters\_host.cpp.

4.2.2.2 void compute\_omp ( float \* *h\_in*[], float \* *h\_reference*[], gpu\_arrays \* *gpuarrays*, params \* *gparams*, cmd\_args \* *args*, filter\_arrays \* *farr*, int *N* )

Execution of the multirate filter configuration with OpenMP.

Parameters

<i>h_in</i>	array containing all input blocks of the MR filter
<i>h_reference</i>	array containing the MR filter output
<i>gpuarrays</i>	structure containing the gpu arrays and parameters
<i>gparams</i>	structure containing parameters
<i>args</i>	structure containing command line arguments
<i>farr</i>	structure containing the filters
<i>N</i>	the number of input blocks to be processed.

Definition at line 123 of file filters\_host.cpp.

4.2.2.3 void compute\_ref ( float \* *h\_in*[], float \* *h\_reference*[], gpu\_arrays \* *gpuarrays*, params \* *gparams*, cmd\_args \* *args*, filter\_arrays \* *farr*, int *N* )

Execution of the multirate filter configuration serially.

## Parameters

<i>h_in</i>	array containing all input blocks of the MR filter
<i>h_reference</i>	array containing the MR filter output
<i>gpuarrays</i>	structure containing the gpu arrays and parameters
<i>gparams</i>	structure containing parameters
<i>args</i>	structure containing command line arguments
<i>farr</i>	structure containing the filters
<i>N</i>	the number of input blocks to be processed.

Definition at line 69 of file filters\_host.cpp.

## 4.2.2.4 void print\_usage ( )

Print command line usage.

Definition at line 314 of file filters\_host.cpp.

## 4.2.2.5 void read\_command\_line ( int argc, char \* argv[], cmd\_args \* args )

Read arguments from command line

Definition at line 231 of file filters\_host.cpp.

## 4.2.3 Variable Documentation

## 4.2.3.1 int rdiv[MAX\_RATES] = {2, 10, 50, 4, 16, 32, 8, 24, 36, 42} [static]

Decimation factors for multiple input rates.

Definition at line 75 of file filters.h.

## 4.3 filters\_cuda.cu File Reference

```
#include "filters.h"
#include <helper_cuda.h>
```

## Functions

- `__global__ void multiFilterGpuDevice_2dgrid` (float \*in, float \*out, float \*filter, int osize, int isize, int bsize)
- `__global__ void multiFilterGpuDevice_2dgrid_shmem` (float \*in, float \*out, float \*filter, int osize, int isize, int bsize)
- void `cudaMultiFilterFirInit` (float \*\*in, float \*\*out, float \*filters, params \*params, gpu\_arrays \*arrays)
- void `cudaMultiFilterFirClose` (params \*params, gpu\_arrays \*arrays)
- void `cudaMultiFilterFirStreams` (gpu\_arrays \*arrays, params \*params)

## 4.3.1 Function Documentation

## 4.3.1.1 void cudaMultiFilterFirClose ( params \* params, gpu\_arrays \* arrays )

Clean up CUDA resources

## Parameters

<i>params</i>	structure for the filters parameters, which contains the input and output sizes for each filter as well as the number of filters and sampling rates to be processed.
<i>arrays</i>	a structure of pointers to the GPU arrays for reuse

Definition at line 220 of file filters\_cuda.cu.

#### 4.3.1.2 void cudaMultiFilterFirInit ( float \*\* in, float \*\* out, float \* filters, params \* params, gpu\_arrays \* arrays )

Perform all memory allocations and transfers required for a GPU multiple filter operation. This also allows to keep allocated memory and reusable data on the GPU, in order to repeat CUDA calls along the whole input signal.

## Parameters

<i>in</i>	the input buffers (2D array), for resampled inputs
<i>out</i>	the output buffers (2D array), different sampling
<i>filters</i>	a two dimensional array containing the coefficients for multiple filters
<i>params</i>	structure for the filters parameters, which contains the input and output sizes for each filter as well as the number of filters and sampling rates to be processed.
<i>arrays</i>	a structure of pointers to the GPU arrays for reuse

Definition at line 139 of file filters\_cuda.cu.

#### 4.3.1.3 void cudaMultiFilterFirStreams ( gpu\_arrays \* arrays, params \* params )

This function performs multiple filters with multiple input sampling rates on a GPU. The required memory is pre-allocated on the GPU and filters also copied in advance. The different rates are executed asynchronously on the GPU using CUDA streams.

## Parameters

<i>arrays</i>	all the GPU and host pinned memory input and output arrays we need for asynchronous operation
<i>params</i>	structure for the filters parameters, which contains the input and output sizes for each filter as well as the number of filters and sampling rates to be processed.

Definition at line 248 of file filters\_cuda.cu.

#### 4.3.1.4 \_\_global\_\_ void multiFilterGpuDevice\_2dgrid ( float \* in, float \* out, float \* filter, int osize, int isize, int bsize )

This GPU kernel applies multiple filters, one per y direction. It is assumed that input data arrays are contiguous in memory.

Definition at line 23 of file filters\_cuda.cu.

#### 4.3.1.5 \_\_global\_\_ void multiFilterGpuDevice\_2dgrid\_shmem ( float \* in, float \* out, float \* filter, int osize, int isize, int bsize )

This kernel applies multiple filters, one per y direction. It is assumed that input data arrays are contiguous in memory.

Definition at line 81 of file filters\_cuda.cu.

## 4.4 filters\_host.cpp File Reference

```
#include <iostream>
```

```
#include <stdio>
#include <stdlib>
#include <cmath>
#include <getopt.h>
#include <sys/time.h>
#include <omp.h>
#include "filters.h"
```

## Functions

- void [filterCpuFir](#) (float \*out, float \*in, [filter\\_arrays](#) \*farr, int n, int nf)
- void [compute\\_ref](#) (float \*h\_in[], float \*h\_reference[], [gpu\\_arrays](#) \*gpuarrays, [params](#) \*gparams, [cmd\\_args](#) \*args, [filter\\_arrays](#) \*farr, int N)
- void [compute\\_omp](#) (float \*h\_in[], float \*h\_reference[], [gpu\\_arrays](#) \*gpuarrays, [params](#) \*gparams, [cmd\\_args](#) \*args, [filter\\_arrays](#) \*farr, int N)
- void [check\\_results](#) (float \*h\_reference[], float \*h\_out[], [gpu\\_arrays](#) \*gpuarrays, [params](#) \*gparams, int N)
- void [read\\_command\\_line](#) (int argc, char \*argv[], [cmd\\_args](#) \*args)
- void [print\\_usage](#) ()

### 4.4.1 Function Documentation

#### 4.4.1.1 void [check\\_results](#) ( float \* [h\\_reference](#)[], float \* [h\\_out](#)[], [gpu\\_arrays](#) \* [gpuarrays](#), [params](#) \* [gparams](#), int *N* )

Check the relative error between CPU and GPU filter computation results

##### Parameters

<i>h_in</i>	array containing all input blocks of the MR filter
<i>h_reference</i>	array containing the MR filter CPU output
<i>h_out</i>	array containing the MR filter GPU output
<i>gpuarrays</i>	structure containing the gpu arrays and parameters
<i>gparams</i>	structure containing parameters
<i>N</i>	the number of input blocks to be processed.

Definition at line 188 of file `filters_host.cpp`.

#### 4.4.1.2 void [compute\\_omp](#) ( float \* [h\\_in](#)[], float \* [h\\_reference](#)[], [gpu\\_arrays](#) \* [gpuarrays](#), [params](#) \* [gparams](#), [cmd\\_args](#) \* [args](#), [filter\\_arrays](#) \* [farr](#), int *N* )

Execution of the multirate filter configuration with OpenMP.

##### Parameters

<i>h_in</i>	array containing all input blocks of the MR filter
<i>h_reference</i>	array containing the MR filter output
<i>gpuarrays</i>	structure containing the gpu arrays and parameters
<i>gparams</i>	structure containing parameters
<i>args</i>	structure containing command line arguments
<i>farr</i>	structure containing the filters
<i>N</i>	the number of input blocks to be processed.

Definition at line 123 of file `filters_host.cpp`.

4.4.1.3 void compute\_ref ( float \* *h\_in*[], float \* *h\_reference*[], gpu\_arrays \* *gpuarrays*, params \* *gparams*, cmd\_args \* *args*, filter\_arrays \* *farr*, int *N* )

Execution of the multirate filter configuration serially.

## Parameters

<i>h_in</i>	array containing all input blocks of the MR filter
<i>h_reference</i>	array containing the MR filter output
<i>gpuarrays</i>	structure containing the gpu arrays and parameters
<i>gparams</i>	structure containing parameters
<i>args</i>	structure containing command line arguments
<i>farr</i>	structure containing the filters
<i>N</i>	the number of input blocks to be processed.

Definition at line 69 of file filters\_host.cpp.

#### 4.4.1.4 void filterCpuFir ( float \* out, float \* in, filter\_arrays \* farr, int n, int nf )

Basic serial implementation of an FIR filter. Based on the Tipic Vamp plugin .

## Parameters

<i>out</i>	the output array from a single filter
<i>in</i>	the input array
<i>farr</i>	structure containing filter object
<i>n</i>	length of input/output
<i>nf</i>	the number of the filter being processed

Definition at line 30 of file filters\_host.cpp.

#### 4.4.1.5 void print\_usage ( )

Print command line usage.

Definition at line 314 of file filters\_host.cpp.

#### 4.4.1.6 void read\_command\_line ( int argc, char \* argv[], cmd\_args \* args )

Read arguments from command line

Definition at line 231 of file filters\_host.cpp.



# Index

A\_SIZE  
filters.h, 12

B\_SIZE  
filters.h, 12

bk  
filter\_arrays, 6  
buf\_in  
filter\_arrays, 6  
buf\_out  
filter\_arrays, 6

CUDA\_SHM  
filters.h, 12

check\_results  
filters.h, 13  
filters\_host.cpp, 16

cmd\_args, 5  
insize, 5  
nf, 5  
nrates, 5  
rconst, 6  
tim, 6

compute\_omp  
filters.h, 13  
filters\_host.cpp, 16

compute\_ref  
filters.h, 13  
filters\_host.cpp, 16

cudaMultiFilterFirClose  
filters\_cuda.cu, 14

cudaMultiFilterFirInit  
filters\_cuda.cu, 15

cudaMultiFilterFirStreams  
filters\_cuda.cu, 15

d\_filters  
gpu\_arrays, 7

d\_in  
gpu\_arrays, 7

d\_out  
gpu\_arrays, 8

filter\_arrays, 6  
bk, 6  
buf\_in, 6  
buf\_out, 6  
m\_offb, 7  
filterCpuFir  
filters\_host.cpp, 18

filtermain.cpp, 11  
main, 11

filters.h, 11  
A\_SIZE, 12  
B\_SIZE, 12  
CUDA\_SHM, 12  
check\_results, 13  
compute\_omp, 13  
compute\_ref, 13  
MAX\_FILTERS, 12  
MAX\_RATES, 12  
OFFSET, 13  
print\_usage, 14  
rdiv, 14  
read\_command\_line, 14  
SIZE\_MUL, 13

filters\_cuda.cu, 14  
cudaMultiFilterFirClose, 14  
cudaMultiFilterFirInit, 15  
cudaMultiFilterFirStreams, 15  
multiFilterGpuDevice\_2dgrid, 15  
multiFilterGpuDevice\_2dgrid\_shmem, 15

filters\_host.cpp, 15  
check\_results, 16  
compute\_omp, 16  
compute\_ref, 16  
filterCpuFir, 18  
print\_usage, 18  
read\_command\_line, 18

fsize  
params, 9

gpu\_arrays, 7  
d\_filters, 7  
d\_in, 7  
d\_out, 8  
h\_in, 8  
h\_out, 8  
isize, 8  
osize, 8  
stream, 8

h\_in  
gpu\_arrays, 8

h\_out  
gpu\_arrays, 8

insize  
cmd\_args, 5

isize

- gpu\_arrays, 8
- m\_offb
  - filter\_arrays, 7
- MAX\_FILTERS
  - filters.h, 12
- MAX\_RATES
  - filters.h, 12
- main
  - filtermain.cpp, 11
- multiFilterGpuDevice\_2dgrid
  - filters\_cuda.cu, 15
- multiFilterGpuDevice\_2dgrid\_shmem
  - filters\_cuda.cu, 15
- nf
  - cmd\_args, 5
- nfilters
  - params, 9
- nrates
  - cmd\_args, 5
  - params, 9
- OFFSET
  - filters.h, 13
- osize
  - gpu\_arrays, 8
- params, 8
  - fsize, 9
  - nfilters, 9
  - nrates, 9
  - rnumf, 9
  - streams, 9
- print\_usage
  - filters.h, 14
  - filters\_host.cpp, 18
- rconst
  - cmd\_args, 6
- rdiv
  - filters.h, 14
- read\_command\_line
  - filters.h, 14
  - filters\_host.cpp, 18
- rnumf
  - params, 9
- SIZE\_MUL
  - filters.h, 13
- stream
  - gpu\_arrays, 8
- streams
  - params, 9
- tim
  - cmd\_args, 6