

# FAST-GPU

1.0

Generated by Doxygen 1.8.15



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 corner Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Function Documentation	5
3.1.2.1 operator()	5
3.1.3 Member Data Documentation	6
3.1.3.1 score	6
3.1.3.2 x	6
3.1.3.3 y	6
<b>4 File Documentation</b>	<b>7</b>
4.1 FAST-GPU/cuda.cu File Reference	7
4.1.1 Function Documentation	7
4.1.1.1 comparator()	8
4.1.1.2 complex_test()	8
4.1.1.3 coords_2to1()	9
4.1.1.4 FAST_global()	9
4.1.1.5 FAST_shared()	10
4.1.1.6 fast_test()	11
4.1.1.7 fill_const_mem()	11
4.1.1.8 find_corners()	12
4.1.1.9 get_score()	12
4.2 FAST-GPU/cuda.cuh File Reference	13
4.2.1 Detailed Description	14
4.2.2 Macro Definition Documentation	14
4.2.2.1 BLOCK_SIZE	14
4.2.2.2 CHECK_ERROR	14
4.2.2.3 CIRCLE_SIZE	15
4.2.2.4 CUDA_H	15
4.2.2.5 MASK_SIZE	15
4.2.2.6 PADDING	15
4.2.3 Typedef Documentation	15
4.2.3.1 corner	15
4.2.4 Function Documentation	15
4.2.4.1 complex_test()	15
4.2.4.2 FAST_global()	16
4.2.4.3 FAST_shared()	17

4.2.4.4 fast_test()	18
4.2.4.5 fill_const_mem()	19
4.2.4.6 find_corners()	19
4.2.5 Variable Documentation	20
4.2.5.1 d_circle	20
4.2.5.2 d_mask	20
4.2.5.3 d_mask_shared	20
4.3 FAST-GPU/FAST.cpp File Reference	20
4.3.1 Function Documentation	21
4.3.1.1 allocate_new_image()	21
4.3.1.2 cpu_FAST()	22
4.3.1.3 create_circle()	22
4.3.1.4 create_mask()	23
4.3.1.5 fill_gpu_const_mem()	23
4.3.1.6 free_all_memory()	23
4.3.1.7 init_gpu()	24
4.3.1.8 main()	24
4.3.1.9 obtain_sorted_results()	24
4.3.1.10 parse_args()	25
4.3.1.11 print_device_array()	25
4.3.1.12 run_fast_algo()	27
4.3.1.13 run_on_cpu()	27
4.3.1.14 show_image()	28
4.3.1.15 write_circles()	28
4.4 FAST-GPU/FAST.hpp File Reference	28
4.4.1 Detailed Description	29
4.4.2 Variable Documentation	29
4.4.2.1 circle_size	29
4.4.2.2 end	30
4.4.2.3 filename	30
4.4.2.4 foto	30
4.4.2.5 h_circle	30
4.4.2.6 h_corner_bools	30
4.4.2.7 h_img	30
4.4.2.8 h_mask	31
4.4.2.9 h_mask_shared	31
4.4.2.10 memory_s	31
4.4.2.11 mode	31
4.4.2.12 pi	31
4.4.2.13 start	31
4.4.2.14 threshold	32
4.4.2.15 time_measured	32

---

4.4.2.16 video . . . . .	32
4.4.2.17 work_s . . . . .	32
<b>Index</b>	<b>33</b>



# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">corner</a>	Structure holding info about corner . . . . .	<a href="#">5</a>
------------------------	---	-------------------





## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

FAST-GPU/ <a href="#">cuda.cu</a> . . . . .	7
FAST-GPU/ <a href="#">cuda.cuh</a>	
This header exposes all kernels and important methods for CUDA invocation . . . . .	13
FAST-GPU/ <a href="#">FAST.cpp</a> . . . . .	20
FAST-GPU/ <a href="#">FAST.hpp</a>	
Main header for this program. It has all important includes and global variables . . . . .	28



## Chapter 3

# Class Documentation

### 3.1 corner Struct Reference

Structure holding info about corner.

#### Public Member Functions

- `__host__ __device__ bool operator() (const corner &c1, const corner &c2)`

#### Public Attributes

- unsigned [score](#)
- unsigned [x](#)
- unsigned [y](#)

#### 3.1.1 Detailed Description

Structure holding info about corner.

Definition at line 46 of file `cuda.cuh`.

#### 3.1.2 Member Function Documentation

##### 3.1.2.1 operator()

```
__host__ __device__ bool corner::operator() (  
    const corner & c1,  
    const corner & c2 ) [inline]
```

Definition at line 51 of file `cuda.cuh`.

### 3.1.3 Member Data Documentation

#### 3.1.3.1 score

`unsigned corner::score`

Definition at line 47 of file `cuda.cuh`.

#### 3.1.3.2 x

`unsigned corner::x`

Definition at line 48 of file `cuda.cuh`.

#### 3.1.3.3 y

`unsigned corner::y`

Definition at line 49 of file `cuda.cuh`.

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

- FAST-GPU/[cuda.cuh](#)

# Chapter 4

## File Documentation

### 4.1 FAST-GPU/cuda.cu File Reference

```
#include "cuda.cuh"
```

#### Functions

- `__device__ __host__ char comparator` (unsigned char pixel\_val, unsigned char circle\_val, int threshold, char sign)  
*Comparator using threshold.*
- `__device__ __host__ int get_score` (int pixel\_val, int circle\_val, int threshold)  
*Calculate element of score of given pixel.*
- `__device__ int coords_2to1` (int x, int y, int width, int height, bool eliminate\_padding)  
*Recalculate 2D indexing into 1D.*
- `__host__ void fill_const_mem` (int \*h\_circle, int \*h\_mask, int \*h\_mask\_shared)  
*Loads circle and mask from host to device constant memory.*
- `__device__ __host__ char fast_test` (unsigned char \*input, int \*circle, int threshold, int id)  
*Perform fast test on pixel with given id.*
- `__device__ __host__ int complex_test` (unsigned char \*input, unsigned \*scores, unsigned \*corner\_bools, int \*circle, int threshold, int pi, int s\_id, int g\_id)  
*Run complex test on pixel with given id.*
- `__global__ void FAST_global` (unsigned char \*input, unsigned \*scores, unsigned \*corner\_bools, int width, int height, int threshold, int pi)  
*Kernel computing FAST algorithm using global memory.*
- `__global__ void FAST_shared` (unsigned char \*input, unsigned \*scores, unsigned \*corner\_bools, int width, int height, int threshold, int pi)  
*Kernel computing FAST algorithm using shared memory.*
- `__global__ void find_corners` (unsigned \*scanned\_array, corner \*result, unsigned \*scores, int length, int width)  
*Kernel to obtain array of corners from scanned array.*

#### 4.1.1 Function Documentation

#### 4.1.1.1 comparator()

```
__device__ __host__ char comparator (
    unsigned char pixel_val,
    unsigned char circle_val,
    int threshold,
    char sign )
```

Comparator using threshold.

##### Parameters

<i>pixel_val</i>	value of center pixel
<i>circle_val</i>	value of pixel in circle
<i>threshold</i>	
<i>sign</i>	modifies function of comparator

##### Returns

char boolean

return boolean if true ... sign parameter gives us criterion

Definition at line 13 of file cuda.cu.

#### 4.1.1.2 complex\_test()

```
__device__ __host__ int complex_test (
    unsigned char * input,
    unsigned * scores,
    unsigned * corner_bools,
    int * circle,
    int threshold,
    int pi,
    int s_id,
    int g_id )
```

Run complex test on pixel with given id.

##### Parameters

<i>input</i>	image array
<i>scores</i>	array to output score
<i>corner_bools</i>	array to output whether pixel is corner or not
<i>circle</i>	
<i>threshold</i>	
<i>pi</i>	
<i>s_id</i>	1D index in shared memory (same as <i>g_id</i> when using only global memory)
<i>g_id</i>	1D index in global memory

**Returns**

int score of pixel with given id

make complex test and calculate score

iterate over whole circle

signum

Definition at line 124 of file cuda.cu.

**4.1.1.3 coords\_2to1()**

```
__device__ int coords_2to1 (
    int x,
    int y,
    int width,
    int height,
    bool eliminate_padding )
```

Recalculate 2D indexing into 1D.

**Parameters**

<i>x</i>	
<i>y</i>	
<i>width</i>	width of image
<i>height</i>	height of image
<i>eliminate_padding</i>	boolean telling whether to eliminate borders of image

**Returns**

int element of score

cutout the borders of image, only active when eliminate\_padding == true

Definition at line 58 of file cuda.cu.

**4.1.1.4 FAST\_global()**

```
__global__ void FAST_global (
    unsigned char * input,
    unsigned * scores,
    unsigned * corner_bools,
    int width,
    int height,
    int threshold,
    int pi )
```

Kernel computing FAST algorithm using global memory.

kernel methods

**Parameters**

<i>input</i>	image array
<i>scores</i>	array to output score
<i>corner_bools</i>	array to output whether pixel is corner or not
<i>width</i>	width of image
<i>height</i>	height of image
<i>threshold</i>	
<i>pi</i>	

get 1d coordinates and cutout borders

fast test, it turns out that it slows the code a little bit

complex test

non-maximal suppression

Definition at line 180 of file cuda.cu.

**4.1.1.5 FAST\_shared()**

```
__global__ void FAST_shared (
    unsigned char * input,
    unsigned * scores,
    unsigned * corner_bools,
    int width,
    int height,
    int threshold,
    int pi )
```

Kernel computing FAST algorithm using shared memory.

**Parameters**

<i>input</i>	image array
<i>scores</i>	array to output score
<i>corner_bools</i>	array to output whether pixel is corner or not
<i>width</i>	width of image
<i>height</i>	height of image
<i>threshold</i>	
<i>pi</i>	

final score of corner in particular thread

get 1d coordinates and cutout borders

fill in shared memory

fast test



make complex test and calculate score

refill shared memory

non-max suppression

Definition at line 227 of file cuda.cu.

#### 4.1.1.6 fast\_test()

```
__device__ __host__ char fast_test (
    unsigned char * input,
    int * circle,
    int threshold,
    int id )
```

Perform fast test on pixel with given id.

##### Parameters

<i>input</i>	image array
<i>circle</i>	
<i>threshold</i>	
<i>id</i>	pixel 1D index

##### Returns

boolean telling whether it is corner candidate

Definition at line 92 of file cuda.cu.

#### 4.1.1.7 fill\_const\_mem()

```
__host__ void fill_const_mem (
    int * h_circle,
    int * h_mask,
    int * h_mask_shared )
```

Loads circle and mask from host to device constant memory.

##### Parameters

<i>h_circle</i>	circle array
<i>h_mask</i>	mask array
<i>h_mask_shared</i>	mask array for shared memory

Definition at line 75 of file cuda.cu.

#### 4.1.1.8 find\_corners()

```
__global__ void find_corners (
    unsigned * scanned_array,
    corner * result,
    unsigned * scores,
    int length,
    int width )
```

Kernel to obtain array of corners from scanned array.

##### Parameters

<i>scanned_array</i>	array which is output of parallel scan over array of booleans
<i>result</i>	output corners
<i>scores</i>	array of scores of all pixels
<i>length</i>	number of pixels in image
<i>width</i>	width of image

Definition at line 302 of file cuda.cu.

#### 4.1.1.9 get\_score()

```
__device__ __host__ int get_score (
    int pixel_val,
    int circle_val,
    int threshold )
```

Calculate element of score of given pixel.

##### Parameters

<i>pixel_val</i>	value of center pixel
<i>circle_val</i>	value of pixel in circle
<i>threshold</i>	

##### Returns

int element of score

returns score of circle element, positive when higher, negative when lower intensity

Definition at line 31 of file cuda.cu.

## 4.2 FAST-GPU/cuda.cuh File Reference

This header exposes all kernels and important methods for CUDA invocation.

```
#include "cuda_runtime.h"
#include "device_launch_parameters.h"
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <time.h>
#include <thrust/scan.h>
#include <thrust/sort.h>
#include <thrust/device_vector.h>
#include <thrust/execution_policy.h>
```

### Classes

- struct [corner](#)  
*Structure holding info about corner.*

### Macros

- #define [CUDA\\_H](#)
- #define [PADDING](#) 3  
*constants*
- #define [BLOCK\\_SIZE](#) 32
- #define [CIRCLE\\_SIZE](#) 16
- #define [MASK\\_SIZE](#) 3
- #define [CHECK\\_ERROR](#)(error) ( HandleError( error, \_\_FILE\_\_, \_\_LINE\_\_ ) )

### Typedefs

- typedef struct [corner](#) [corner](#)  
*Structure holding info about corner.*

### Functions

- `__global__ void FAST\_global (unsigned char *input, unsigned *scores, unsigned *corner_bools, int width, int height, int threshold, int pi)`  
*kernel methods*
- `__global__ void FAST\_shared (unsigned char *input, unsigned *scores, unsigned *corner_bools, int width, int height, int threshold, int pi)`  
*Kernel computing FAST algorithm using shared memory.*
- `__host__ void fill\_const\_mem (int *h\_circle, int *h\_mask, int *h\_mask\_shared)`  
*Loads circle and mask from host to device constant memory.*
- `__global__ void find\_corners (unsigned *scanned_array, corner *result, unsigned *scores, int length, int width)`  
*Kernel to obtain array of corners from scanned array.*
- `__device__ __host__ char fast\_test (unsigned char *input, int *circle, int threshold, int id)`  
*Perform fast test on pixel with given id.*
- `__device__ __host__ int complex\_test (unsigned char *input, unsigned *scores, unsigned *corner_bools, int *circle, int threshold, int pi, int s_id, int g_id)`  
*Run complex test on pixel with given id.*

## Variables

- `__constant__ int d_circle [CIRCLE_SIZE]`
- `__constant__ int d_mask [MASK_SIZE *MASK_SIZE]`
- `__constant__ int d_mask_shared [MASK_SIZE *MASK_SIZE]`

### 4.2.1 Detailed Description

This header exposes all kernels and important methods for CUDA invocation.

#### Author

Zdenek Rozsypalek ( [rozsyzde@fel.cvut.cz](mailto:rozsyzde@fel.cvut.cz) )

#### Version

1.0

#### Date

2019-01-07

#### Copyright

Copyright (c) 2019

### 4.2.2 Macro Definition Documentation

#### 4.2.2.1 BLOCK\_SIZE

```
#define BLOCK_SIZE 32
```

Definition at line 29 of file cuda.cuh.

#### 4.2.2.2 CHECK\_ERROR

```
#define CHECK_ERROR(  
    error ) ( HandleError( error, __FILE__, __LINE__ ) )
```

Definition at line 32 of file cuda.cuh.

#### 4.2.2.3 CIRCLE\_SIZE

```
#define CIRCLE_SIZE 16
```

Definition at line 30 of file cuda.cuh.

#### 4.2.2.4 CUDA\_H

```
#define CUDA_H
```

Definition at line 14 of file cuda.cuh.

#### 4.2.2.5 MASK\_SIZE

```
#define MASK_SIZE 3
```

Definition at line 31 of file cuda.cuh.

#### 4.2.2.6 PADDING

```
#define PADDING 3
```

constants

Definition at line 28 of file cuda.cuh.

### 4.2.3 Typedef Documentation

#### 4.2.3.1 corner

```
typedef struct corner corner
```

Structure holding info about corner.

### 4.2.4 Function Documentation

#### 4.2.4.1 complex\_test()

```
__device__ __host__ int complex_test (  
    unsigned char * input,  
    unsigned * scores,  
    unsigned * corner_bools,  
    int * circle,  
    int threshold,  
    int pi,  
    int s_id,  
    int g_id )
```

Run complex test on pixel with given id.

**Parameters**

<i>input</i>	image array
<i>scores</i>	array to output score
<i>corner_bools</i>	array to output whether pixel is corner or not
<i>circle</i>	
<i>threshold</i>	
<i>pi</i>	
<i>s_id</i>	1D index in shared memory (same as <i>g_id</i> when using only global memory)
<i>g_id</i>	1D index in global memory

**Returns**

int score of pixel with given id

make complex test and calculate score

iterate over whole circle

signum

Definition at line 124 of file cuda.cu.

**4.2.4.2 FAST\_global()**

```
__global__ void FAST_global (
    unsigned char * input,
    unsigned * scores,
    unsigned * corner_bools,
    int width,
    int height,
    int threshold,
    int pi )
```

kernel methods

kernel methods

**Parameters**

<i>input</i>	image array
<i>scores</i>	array to output score
<i>corner_bools</i>	array to output whether pixel is corner or not
<i>width</i>	width of image
<i>height</i>	height of image
<i>threshold</i>	
<i>pi</i>	

get 1d coordinates and cutout borders

fast test, it turns out that it slows the code a little bit

complex test

non-maximal suppression

Definition at line 180 of file cuda.cu.

#### 4.2.4.3 FAST\_shared()

```
__global__ void FAST_shared (
    unsigned char * input,
    unsigned * scores,
    unsigned * corner_bools,
    int width,
    int height,
    int threshold,
    int pi )
```

Kernel computing FAST algorithm using shared memory.

##### Parameters

<i>input</i>	image array
<i>scores</i>	array to output score
<i>corner_bools</i>	array to output whether pixel is corner or not
<i>width</i>	width of image
<i>height</i>	height of image
<i>threshold</i>	
<i>pi</i>	

final score of corner in particular thread

get 1d coordinates and cutout borders

fill in shared memory

fast test

make complex test and calculate score

refill shared memory

non-max suppression

Definition at line 227 of file cuda.cu.

#### 4.2.4.4 `fast_test()`

```
__device__ __host__ char fast_test (
    unsigned char * input,
    int * circle,
    int threshold,
    int id )
```

Perform fast test on pixel with given id.



## Parameters

<i>input</i>	image array
<i>circle</i>	
<i>threshold</i>	
<i>id</i>	pixel 1D index

## Returns

boolean telling whether it is corner candidate

Definition at line 92 of file cuda.cu.

## 4.2.4.5 fill\_const\_mem()

```
__host__ void fill_const_mem (
    int * h_circle,
    int * h_mask,
    int * h_mask_shared )
```

Loads circle and mask from host to device constant memory.

## Parameters

<i>h_circle</i>	circle array
<i>h_mask</i>	mask array
<i>h_mask_shared</i>	mask array for shared memory

Definition at line 75 of file cuda.cu.

## 4.2.4.6 find\_corners()

```
__global__ void find_corners (
    unsigned * scanned_array,
    corner * result,
    unsigned * scores,
    int length,
    int width )
```

Kernel to obtain array of corners from scanned array.

## Parameters

<i>scanned_array</i>	array which is output of parallel scan over array of booleans
<i>result</i>	output corners
<i>scores</i>	array of scores of all pixels
<i>length</i>	number of pixels in image
<i>width</i>	width of image

Definition at line 302 of file cuda.cu.

## 4.2.5 Variable Documentation

### 4.2.5.1 d\_circle

```
__constant__ int d_circle[CIRCLE_SIZE]
```

Definition at line 63 of file cuda.cuh.

### 4.2.5.2 d\_mask

```
__constant__ int d_mask[MASK_SIZE *MASK_SIZE]
```

Definition at line 64 of file cuda.cuh.

### 4.2.5.3 d\_mask\_shared

```
__constant__ int d_mask_shared[MASK_SIZE *MASK_SIZE]
```

Definition at line 65 of file cuda.cuh.

## 4.3 FAST-GPU/FAST.cpp File Reference

```
#include "FAST.hpp"
```

## Functions

- void [show\\_image](#) (cv::Mat img)
- void [print\\_device\\_array](#) (unsigned int \*device\_arr, int length)  
*Method for debugging, prints array on device.*
- void [create\\_circle](#) (int \*circle, int w)  
*Generate 16 incremental indexes of pixels in surrounding circle.*
- void [create\\_mask](#) (int \*mask, int w)  
*Generate incremental indexes of mask used in non-maximal suppression.*
- std::vector< [corner](#) > [cpu\\_FAST](#) (unsigned char \*input, unsigned \*scores, int \*mask, int \*circle, int width, int height)  
*Naive CPU implementation of FAST algorithm.*
- void [parse\\_args](#) (int argc, char \*\*argv)  
*Parsing of main arguments.*
- void [fill\\_gpu\\_const\\_mem](#) (int width, int shared\_width)  
*Fill constant gpu memory with mask and circle arrays.*
- void [init\\_gpu](#) (cv::Mat image, int length, int shared\_width)  
*Initialize GPU memory.*
- void [allocate\\_new\\_image](#) (cv::Mat image, int length, cudaStream\_t stream)  
*Transfer image to CUDA device.*
- void [run\\_fast\\_algo](#) (cv::Mat image, int shared\_width, int length, cudaStream\_t work)  
*Method encapsulating FAST algorithm on GPU.*
- void [write\\_circles](#) (cv::Mat image, [corner](#) \*corners, int number\_of\_corners)  
*Draw circles for all corners (with different color based on their score)*
- [corner](#) \* [obtain\\_sorted\\_results](#) (int length, int \*corners\_num, cudaStream\_t stream, int width)  
*Obtain array of corner structures from array of booleans.*
- void [free\\_all\\_memory](#) ()  
*Free all allocated memory.*
- void [run\\_on\\_cpu](#) (cv::Mat image)  
*Method encapsulating FAST algorithm running on CPU.*
- int [main](#) (int argc, char \*\*argv)

### 4.3.1 Function Documentation

#### 4.3.1.1 [allocate\\_new\\_image\(\)](#)

```
void allocate_new_image (
    cv::Mat image,
    int length,
    cudaStream_t stream )
```

Transfer image to CUDA device.

#### Parameters

<i>image</i>	
<i>length</i>	number of pixels in image
<i>stream</i>	CUDA stream for async memory copy

create array from image and copy image to device

Definition at line 241 of file FAST.cpp.

#### 4.3.1.2 cpu\_FAST()

```
std::vector<corner> cpu_FAST (
    unsigned char * input,
    unsigned * scores,
    int * mask,
    int * circle,
    int width,
    int height )
```

Naive CPU implementation of FAST algorithm.

##### Parameters

<i>input</i>	image in 1D array
<i>scores</i>	helper array caching scores
<i>mask</i>	
<i>circle</i>	
<i>width</i>	width of input
<i>height</i>	height of input

##### Returns

std::vector<corner> vector of found corners

fast test

complex test

non-max suppression

Definition at line 88 of file FAST.cpp.

#### 4.3.1.3 create\_circle()

```
void create_circle (
    int * circle,
    int w )
```

Generate 16 incremental indexes of pixels in surrounding circle.

## Parameters

<i>circle</i>	output array
<i>w</i>	width of data block (i.e. image width or shared mem width)

Definition at line 34 of file FAST.cpp.

#### 4.3.1.4 create\_mask()

```
void create_mask (
    int * mask,
    int w )
```

Generate incremental indexes of mask used in non-maximal suppression.

## Parameters

<i>mask</i>	output array
<i>w</i>	width of data block (i.e. image width or shared mem width)

Definition at line 62 of file FAST.cpp.

#### 4.3.1.5 fill\_gpu\_const\_mem()

```
void fill_gpu_const_mem (
    int width,
    int shared_width )
```

Fill constant gpu memory with mask and circle arrays.

## Parameters

<i>width</i>	width of image
<i>shared_width</i>	width of shared memory

create circle and mask and copy to device

Definition at line 190 of file FAST.cpp.

#### 4.3.1.6 free\_all\_memory()

```
void free_all_memory ( )
```

Free all allocated memory.

free all memory

Definition at line 344 of file FAST.cpp.

#### 4.3.1.7 init\_gpu()

```
void init_gpu (
    cv::Mat image,
    int length,
    int shared_width )
```

Initialize GPU memory.

##### Parameters

<i>image</i>	
<i>length</i>	number of pixels in image
<i>shared_width</i>	width of shared memory

allocate memory

Definition at line 215 of file FAST.cpp.

#### 4.3.1.8 main()

```
int main (
    int argc,
    char ** argv )
```

load image

Capture frame-by-frame

CPU

GPU

swap pointers and allocate new frame

Definition at line 391 of file FAST.cpp.

#### 4.3.1.9 obtain\_sorted\_results()

```
corner* obtain_sorted_results (
    int length,
    int * corners_num,
    cudaStream_t stream,
    int width )
```

Obtain array of corner structures from array of booleans.

**Parameters**

<i>length</i>	number of pixels in image
<i>corners_num</i>	output number of corners here
<i>stream</i>	CUDA stream for async computing
<i>width</i>	width of image

**Returns**

corner\* array of corners

create new CUDA array of corners with appropriate length

scanned values

cast pointer

get number of corners from device

allocate array for results

find results, sort and transfer to host

cast pointer

Definition at line 306 of file FAST.cpp.

**4.3.1.10 parse\_args()**

```
void parse_args (
    int argc,
    char ** argv )
```

Parsing of main arguments.

**Parameters**

<i>argc</i>	
<i>argv</i>	

Definition at line 148 of file FAST.cpp.

**4.3.1.11 print\_device\_array()**

```
void print_device_array (
    unsigned int * device_arr,
    int length )
```

Method for debugging, prints array on device.



## Parameters

<i>device_arr</i>	array to print
<i>length</i>	length of array

Definition at line 15 of file FAST.cpp.

## 4.3.1.12 run\_fast\_algo()

```
void run_fast_algo (
    cv::Mat image,
    int shared_width,
    int length,
    cudaStream_t work )
```

Method encapsulating FAST algorithm on GPU.

## Parameters

<i>image</i>	
<i>shared_width</i>	width of shared memory
<i>length</i>	number of pixels in image
<i>work</i>	CUDA stream for async computing

define grid and block sizes

run kernel and measure the time

run kernel and measure the time

Definition at line 256 of file FAST.cpp.

## 4.3.1.13 run\_on\_cpu()

```
void run_on_cpu (
    cv::Mat image )
```

Method encapsulating FAST algorithm running on CPU.

## Parameters

<i>image</i>	
--------------	--

Definition at line 360 of file FAST.cpp.

#### 4.3.1.14 show\_image()

```
void show_image (
    cv::Mat img )
```

Definition at line 3 of file FAST.cpp.

#### 4.3.1.15 write\_circles()

```
void write_circles (
    cv::Mat image,
    corner * corners,
    int number_of_corners )
```

Draw circles for all corners (with different color based on their score)

##### Parameters

<i>image</i>	
<i>corners</i>	found corners
<i>number_of_corners</i>	

draw corners

Definition at line 284 of file FAST.cpp.

## 4.4 FAST-GPU/FAST.hpp File Reference

Main header for this program. It has all important includes and global variables.

```
#include "cuda.cuh"
#include "opencv2/imgcodecs/imgcodecs.hpp"
#include <opencv2/core.hpp>
#include <opencv2/imgcodecs.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/opencv.hpp>
#include <string>
#include <iostream>
#include <vector>
```

### Variables

- int **threshold** = 75  
*argument parsing*
- int **mode** = 1
- int **pi** = 12

- char \* `filename` = NULL
- bool `video` = false
- bool `foto` = false
- int `circle_size` = 5
- unsigned char \* `h_img`
  - host variables*
- unsigned \* `h_corner_bools`
- int \* `h_circle`
- int \* `h_mask`
- int \* `h_mask_shared`
- cudaStream\_t `memory_s`
  - streams for gpu video*
- cudaStream\_t `work_s`
- clock\_t `start`
  - time measurement*
- clock\_t `end`
- double `time_measured`

#### 4.4.1 Detailed Description

Main header for this program. It has all important includes and global variables.

##### Author

Zdenek Rozsypalek ( [rozsyzde@fel.cvut.cz](mailto:rozsyzde@fel.cvut.cz) )

##### Version

1.0

##### Date

2019-01-07

##### Copyright

Copyright (c) 2019

#### 4.4.2 Variable Documentation

##### 4.4.2.1 `circle_size`

```
int circle_size = 5
```

Definition at line 32 of file FAST.hpp.

#### 4.4.2.2 end

```
clock_t end
```

Definition at line 45 of file FAST.hpp.

#### 4.4.2.3 filename

```
char* filename = NULL
```

Definition at line 29 of file FAST.hpp.

#### 4.4.2.4 foto

```
bool foto = false
```

Definition at line 31 of file FAST.hpp.

#### 4.4.2.5 h\_circle

```
int* h_circle
```

Definition at line 37 of file FAST.hpp.

#### 4.4.2.6 h\_corner\_bools

```
unsigned* h_corner_bools
```

Definition at line 36 of file FAST.hpp.

#### 4.4.2.7 h\_img

```
unsigned char* h_img
```

host variables

Definition at line 35 of file FAST.hpp.

#### 4.4.2.8 h\_mask

```
int* h_mask
```

Definition at line 38 of file FAST.hpp.

#### 4.4.2.9 h\_mask\_shared

```
int* h_mask_shared
```

Definition at line 39 of file FAST.hpp.

#### 4.4.2.10 memory\_s

```
cudaStream_t memory_s
```

streams for gpu video

Definition at line 42 of file FAST.hpp.

#### 4.4.2.11 mode

```
int mode = 1
```

Definition at line 27 of file FAST.hpp.

#### 4.4.2.12 pi

```
int pi = 12
```

Definition at line 28 of file FAST.hpp.

#### 4.4.2.13 start

```
clock_t start
```

time measurement

Definition at line 45 of file FAST.hpp.

#### 4.4.2.14 threshold

```
int threshold = 75
```

argument parsing

Definition at line 26 of file FAST.hpp.

#### 4.4.2.15 time\_measured

```
double time_measured
```

Definition at line 46 of file FAST.hpp.

#### 4.4.2.16 video

```
bool video = false
```

Definition at line 30 of file FAST.hpp.

#### 4.4.2.17 work\_s

```
cudaStream_t work_s
```

Definition at line 42 of file FAST.hpp.

# Index

- allocate\_new\_image
  - FAST.cpp, [21](#)
- BLOCK\_SIZE
  - cuda.cuh, [14](#)
- CHECK\_ERROR
  - cuda.cuh, [14](#)
- CIRCLE\_SIZE
  - cuda.cuh, [14](#)
- circle\_size
  - FAST.hpp, [29](#)
- comparator
  - cuda.cu, [7](#)
- complex\_test
  - cuda.cu, [8](#)
  - cuda.cuh, [15](#)
- coords\_2to1
  - cuda.cu, [9](#)
- corner, [5](#)
  - cuda.cuh, [15](#)
  - operator(), [5](#)
  - score, [6](#)
  - x, [6](#)
  - y, [6](#)
- cpu\_FAST
  - FAST.cpp, [22](#)
- create\_circle
  - FAST.cpp, [22](#)
- create\_mask
  - FAST.cpp, [23](#)
- cuda.cu
  - comparator, [7](#)
  - complex\_test, [8](#)
  - coords\_2to1, [9](#)
  - FAST\_global, [9](#)
  - FAST\_shared, [10](#)
  - fast\_test, [11](#)
  - fill\_const\_mem, [11](#)
  - find\_corners, [12](#)
  - get\_score, [12](#)
- cuda.cuh
  - BLOCK\_SIZE, [14](#)
  - CHECK\_ERROR, [14](#)
  - CIRCLE\_SIZE, [14](#)
  - complex\_test, [15](#)
  - corner, [15](#)
  - CUDA\_H, [15](#)
  - d\_circle, [20](#)
  - d\_mask, [20](#)
  - d\_mask\_shared, [20](#)
  - FAST\_global, [16](#)
  - FAST\_shared, [17](#)
  - fast\_test, [17](#)
  - fill\_const\_mem, [19](#)
  - find\_corners, [19](#)
  - MASK\_SIZE, [15](#)
  - PADDING, [15](#)
- CUDA\_H
  - cuda.cuh, [15](#)
- d\_circle
  - cuda.cuh, [20](#)
- d\_mask
  - cuda.cuh, [20](#)
- d\_mask\_shared
  - cuda.cuh, [20](#)
- end
  - FAST.hpp, [29](#)
- FAST-GPU/cuda.cu, [7](#)
- FAST-GPU/cuda.cuh, [13](#)
- FAST-GPU/FAST.cpp, [20](#)
- FAST-GPU/FAST.hpp, [28](#)
- FAST.cpp
  - allocate\_new\_image, [21](#)
  - cpu\_FAST, [22](#)
  - create\_circle, [22](#)
  - create\_mask, [23](#)
  - fill\_gpu\_const\_mem, [23](#)
  - free\_all\_memory, [23](#)
  - init\_gpu, [24](#)
  - main, [24](#)
  - obtain\_sorted\_results, [24](#)
  - parse\_args, [25](#)
  - print\_device\_array, [25](#)
  - run\_fast\_algo, [27](#)
  - run\_on\_cpu, [27](#)
  - show\_image, [27](#)
  - write\_circles, [28](#)
- FAST.hpp
  - circle\_size, [29](#)
  - end, [29](#)
  - filename, [30](#)
  - foto, [30](#)
  - h\_circle, [30](#)
  - h\_corner\_bools, [30](#)
  - h\_img, [30](#)
  - h\_mask, [30](#)

- h\_mask\_shared, 31
- memory\_s, 31
- mode, 31
- pi, 31
- start, 31
- threshold, 31
- time\_measured, 32
- video, 32
- work\_s, 32
- FAST\_global
  - cuda.cu, 9
  - cuda.cuh, 16
- FAST\_shared
  - cuda.cu, 10
  - cuda.cuh, 17
- fast\_test
  - cuda.cu, 11
  - cuda.cuh, 17
- filename
  - FAST.hpp, 30
- fill\_const\_mem
  - cuda.cu, 11
  - cuda.cuh, 19
- fill\_gpu\_const\_mem
  - FAST.cpp, 23
- find\_corners
  - cuda.cu, 12
  - cuda.cuh, 19
- foto
  - FAST.hpp, 30
- free\_all\_memory
  - FAST.cpp, 23
- get\_score
  - cuda.cu, 12
- h\_circle
  - FAST.hpp, 30
- h\_corner\_bools
  - FAST.hpp, 30
- h\_img
  - FAST.hpp, 30
- h\_mask
  - FAST.hpp, 30
- h\_mask\_shared
  - FAST.hpp, 31
- init\_gpu
  - FAST.cpp, 24
- main
  - FAST.cpp, 24
- MASK\_SIZE
  - cuda.cuh, 15
- memory\_s
  - FAST.hpp, 31
- mode
  - FAST.hpp, 31
- obtain\_sorted\_results
  - FAST.cpp, 24
- operator()
  - corner, 5
- PADDING
  - cuda.cuh, 15
- parse\_args
  - FAST.cpp, 25
- pi
  - FAST.hpp, 31
- print\_device\_array
  - FAST.cpp, 25
- run\_fast\_algo
  - FAST.cpp, 27
- run\_on\_cpu
  - FAST.cpp, 27
- score
  - corner, 6
- show\_image
  - FAST.cpp, 27
- start
  - FAST.hpp, 31
- threshold
  - FAST.hpp, 31
- time\_measured
  - FAST.hpp, 32
- video
  - FAST.hpp, 32
- work\_s
  - FAST.hpp, 32
- write\_circles
  - FAST.cpp, 28
- x
  - corner, 6
- y
  - corner, 6