# SOFTWARE REQUIREMENTS SPECIFICATION

*Face Detection System based on GPGPU Technology*

Version 1.0

Oleh V. Palyanytsya

November 25, 2012

# Table of Contents

# 1. Introduction

## *1.1. Purpose.*

The purpose of this document is to provide a complete description of all the functions and specifications of the Face Detection System based on GPGPU Technology.

The expected audience of this document is the faculty of Applied Mathematics, consumers  who will use this system and the developer.

## *1.2. Scope*

Face Detection System based on GPGPU Technology is designed to quickly detect locations and sizes of human faces in arbitrary digital photos. It detects facial features and ignores anything else, like trees buildings etc.

### *1.3. Definitions, Acronyms, Abbreviations*

| Term | Definition |
|---|---|
| FDS | Face Detection System |
| GPGPU | General Purpose Graphics Processing Unit. |
| CUDA | Compute Unified Device Architecture |
| NVIDIA | An American global technology company based in Santa Clara, California. |
| SIMD | Single instruction, multiple data. Computers with multiple processing elements that perform the same operation on multiple data points simultaneously. |
| GNU | GNU is Not Unix |
| GPL | General Public License |

### *1.4. References*

NVIDIA CUDA Zone - https://developer.nvidia.com/category/zone/cuda-zone

GNU Man Pages Standard - http://www.gnu.org/prep/standards/html_node/Man-Pages.html

“Robust Real-time Object Detection”, Viola, Jones, IJCV, 2001;

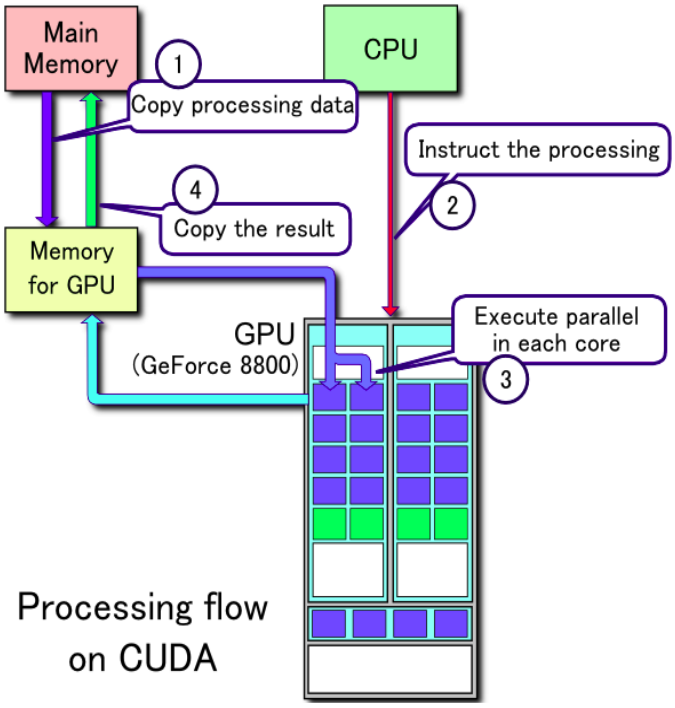
### *1.5. Overview*

Console-based program product developed to quickly detect human faces on digital photos and ignore other objects. Increasing of computation speed is achieved due to using GPGPU technology.

# 2. Overall Description

## *2.1. Product Perspective*

FDS is based on CUDA GPGPU technology. GPGPU is a technology that allows to perform computation in applications traditionally handled by the CPU. CUDA is a parallel computing platform and programming model created by NVIDIA. GPGPU technologies allow to effectively paralyze SIMD tasks and get much more higher performance in comparison with CPU.




**Example of CUDA processing flow**

1. Copy data from main memory to GPU memory
2. CPU instructs the process to GPU
3. GPU execute parallel in each core
4. Copy the result from GPU memory to main memory

### *2.2. Product Functions*

The main function of product is to detect human faces on digital photos.

### *2.3. User Characteristics*

The expected group of users of FDS are companies, that need to detect human faces on digital photos. They may be security companies, companies of video monitoring etc.

The user is expected to be computer literate and to be able to use command prompt.

### *2.4. Constraints*

The FDS must run on the any Linux distribution with 2.6 kernel version. The application is CUDA based, therefore there is need of computer with CUDA compatible graphics card with computing capability 1.1 or higher.

# 3. Specific Requirements

## *3.1. Interface Requirements*

### 3.1.1. User interfaces

FDS provides command line interface (CLI) for user.

Synopsis:

`fds [OPTION]... FILE...`

To start program user must type program name (fds), then specify options (if necessary) and file names.

Options:

`-c` produces only coordinates of rectangle where face is located.

`-i` produces image with drawn rectangle where face is located.

`--color NUMBER NUMBER NUMBER`

allows to choose color of lines if -i option is specified. Otherwise is ignored. The arguments are red, green and blue components of color.

`--thickness NUMBER`

allows to choose thickness of lines if `-i` option is specified. Otherwise is ignored.

`-s` use CUDA streams, if they are supported.

`-b` use AdaBoost.

`--silent` Silent mode. Produce no output, except error messages.

`--silent-no-errrors`

Silent mode. Produce no output and no error messages.

`--log FILE` Specify file for saving all output messages.

`--err FILE` Specify file for saving all error messages.

### 3.1.2. Software interfaces

Product must support universal text interfaces for further extending.

### 3.1.3. Hardware interfaces

The main input device is keyboard.

## *3.2. Functional Requirements*

1. FDS must detect face in digital photo in JPG format. Face must be placed in the photo, so there is ability to locate two eyes, nose and mouth. Face must not take less than 10% of photo's square. The angle of face doesn't matter.
2. FDS must produce the coordinates of rectangle, where face is located: four pairs, starting from upper-left vertex, with X coordinate as first element and Y coordinate as second element.
3. FDS must produce the image with drawn rectangle, where the face is placed. There should be possibility to specify color and thickness of lines.
4. FDS must put all textual output to *stdout* stream, all error messages to *stderr* stream. There should be options to replace standard streams with text files.
5. FDS must provide silent mode with no output or no error messages.
6. FDS must provide option that allows to use CUDA streams if they are supported. If CUDA streams aren't supported, the error message should be produced.
7. FDS must support Adaptive Boosting (AdaBoost) algorithm.

### *3.3. Non-Functional Requirements*

#### 3.3.1. User documentation

The user documentation must be provided in GNU Man Page format. Every option of console interface must be fully documented and have corresponding exhaustive example.

#### 3.3.2. Requirements for system safety

Program product should not affect any OS files. The input files cannot be modified.

#### 3.3.3. Open source

The FDS must be licensed under GNU GPLv3.

#### 3.3.4. Performance

As far as FDS is GPGPU based, it must work faster than implementation on CPU. The GPU and CPU benches are NVIDIA 8200m G and Intel Core 2 Duo 2.6 GHz, correspondingly.

#### 3.3.4. Portability

The platform dependent constructions should not be used in source code for further porting on other operating systems.

The preferred cross-platform libraries are: Boost, Arageli, POCO.

### 3.3.4. Usability

The program should provide GNU Man Page with user documentation. The program should not require superuser rights to work.

## *3.4. Design Constraints*

The FDS must run on the computer with any Linux distribution with 2.6 kernel version and CUDA compatible graphics card with computing capability 1.1 or higher.