

KernelGenius™

User Guide

*Productivity, Performance and Portability
for Computer Vision Programmers*



Author: Thierry LEPLEY

KernelGenius	2013.1	2 of 8
--------------	--------	--------

KernelGenius	2013.1	3 of 8
--------------	--------	--------

TABLE OF CONTENTS

1.	Introduction	5
2.	Package installation and setup.....	6
2.1.	Structure of the KernelGenius package	6
2.2.	Environment setup	6
3.	Using KernelGenius	7
3.1.	KernelGenius compiler execution	7
3.2.	KernelGenius options	7
4.	Some Limitations	8

KernelGenius	2013.1	4 of 8
--------------	--------	--------

Copyright (C) 2013 STMicroelectronics

This tool is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

1. Introduction

KernelGenius is a tool that targets simplifying the programming of computer vision applications on parallel platforms, by allowing describing in a natural way image processing kernels and then building optimized libraries that can be called by the C or C++ main application.

In its current version, KernelGenius generates optimized OpenCL kernels for the STHORM many-core of STMicroelectronics, without requiring any background in parallelization, any knowledge of STHORM architecture and without dedicating the source code to this target. KernelGenius then primarily covers a programmer audience more familiar with Matlab than with low-level programming, who wants the best productivity and performance portability without any or much effort in optimization for STHORM. Users more familiar with parallel computing will also find in KernelGenius a tool to increase their productivity in the computer vision domain.

As depicted in Figure 1, KernelGenius is then one piece of the STHORM programming methodology. Both automatically generated kernels and manually written kernels can interoperate since both being OpenCL kernels. In its current version, KernelGenius focuses on *regular* image processing algorithms and in future version, other applicative cases will be taken into consideration.

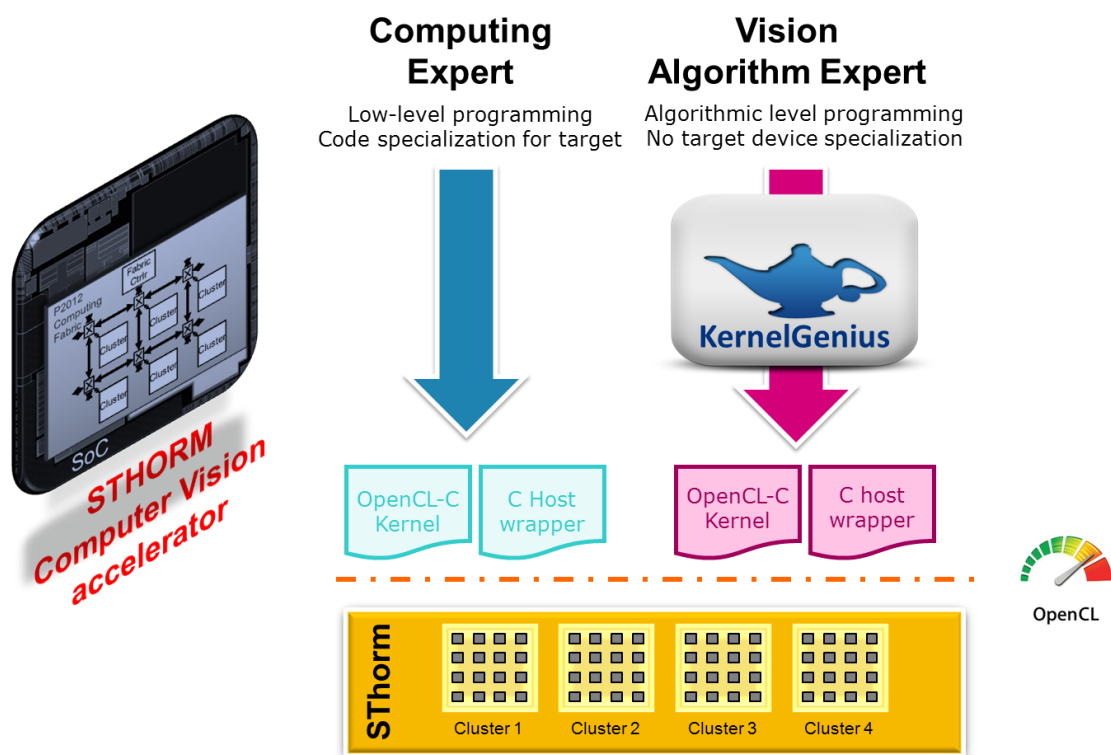


Figure 1 : Programming methodology for STHORM

KernelGenius	2013.1	6 of 8
--------------	--------	--------

2. Package installation and setup

2.1. Structure of the KernelGenius package

For what concerns OpenCL, the STHORM SDK has the following structure:

KernelGenius

```
|-- setup.sh
|-- README.txt
|-- CHANGELOG.txt
|-- bin/
|-- jar/
|-- doc/
|-- target/
|-- examples/
    |-- README
    |-- downsampling/
    |-- gradient/
    |-- imageDifference/
    |-- separableConvolution/
    |-- sobel/
    |-- sobelLowLevel/
    |-- threshold/
    |-- upsampling/
```

- ❖ The `bin` directory contains the driver of the KernelGenius compiler
- ❖ The `target` directory contains internal target description files
- ❖ The `jar` directory contains java libraries of the KernelGenius compiler
- ❖ The `doc` directory contains the KernelGenius documentation
- ❖ The `examples` directory contains a set of relevant applicative examples. In the example directory, a `README` file explains how examples are compiled and executed

2.2. Environment setup

For using the KernelGenius tool, the user must configure its environment by:

- ❖ Using `bash` as shell
- ❖ sourcing the `KernelGenius/setup.sh` file


```
source <install directory path>/setup.sh
```
- ❖ The JVM (Java Virtual Machine) must be installed with `>= 1.6` version

If `JAVA_HOME` is set, KernelGenius will consider it as the java installation directory, otherwise, KernelGenius will consider the '`java`' command of the `PATH` environment variable.

KernelGenius	2013.1	7 of 8
--------------	--------	--------

Examples have a `Makefile` configured to compile and run the OpenCL Code generated by the KernelGenius compiler on the STHORM many-core. Running these examples then require also having installed and setup the STHORM SDK (2013.1 minimal version)

3. Using KernelGenius

3.1. *KernelGenius compiler execution*

The compiler has a driver which handles the preprocessing, the parsing, and the code generation. The command line for executing the KernelGenius compiler is the following:

```
kgenc [option]* <source file>+
```

3.2. *KernelGenius options*

User options:

- `-v` : display compiler version
- `--report` : display a code generation report (memory usage)
- `--verbose <level>` : display more warnings for application developer
 - 1: display additional warnings
- `--help` : help
- `-E` : stop the compilation process after the preprocessing
- `--keep` : keep intermediate files
- `--outdir <name>` : specifies the output directory of generated files
- `-o <name>` : name of the generated OpenCL program.

By default, the name of the OpenCL generated program is the name of the source file to which is removed the `.kg` suffix

Code generation options:

- `--async` : Merge the kernel graph in async mode, resulting in a better load balancing, but a larger memory requirement of the generated OpenCL program.

KernelGenius	2013.1	8 of 8
--------------	--------	--------

4. Some Limitations

- ❖ The reporting of syntax errors are in the current version relatively simple and may not be very explicit in all cases
- ❖ KernelGenius is creating temporary directory to handle temporary files. In some rare cases like syntax of errors in the input file, the compiler may not delete the temporary directory. In this case the user will have to delete it manually (directory starting with '_KG')