

Jetson で OpenCV

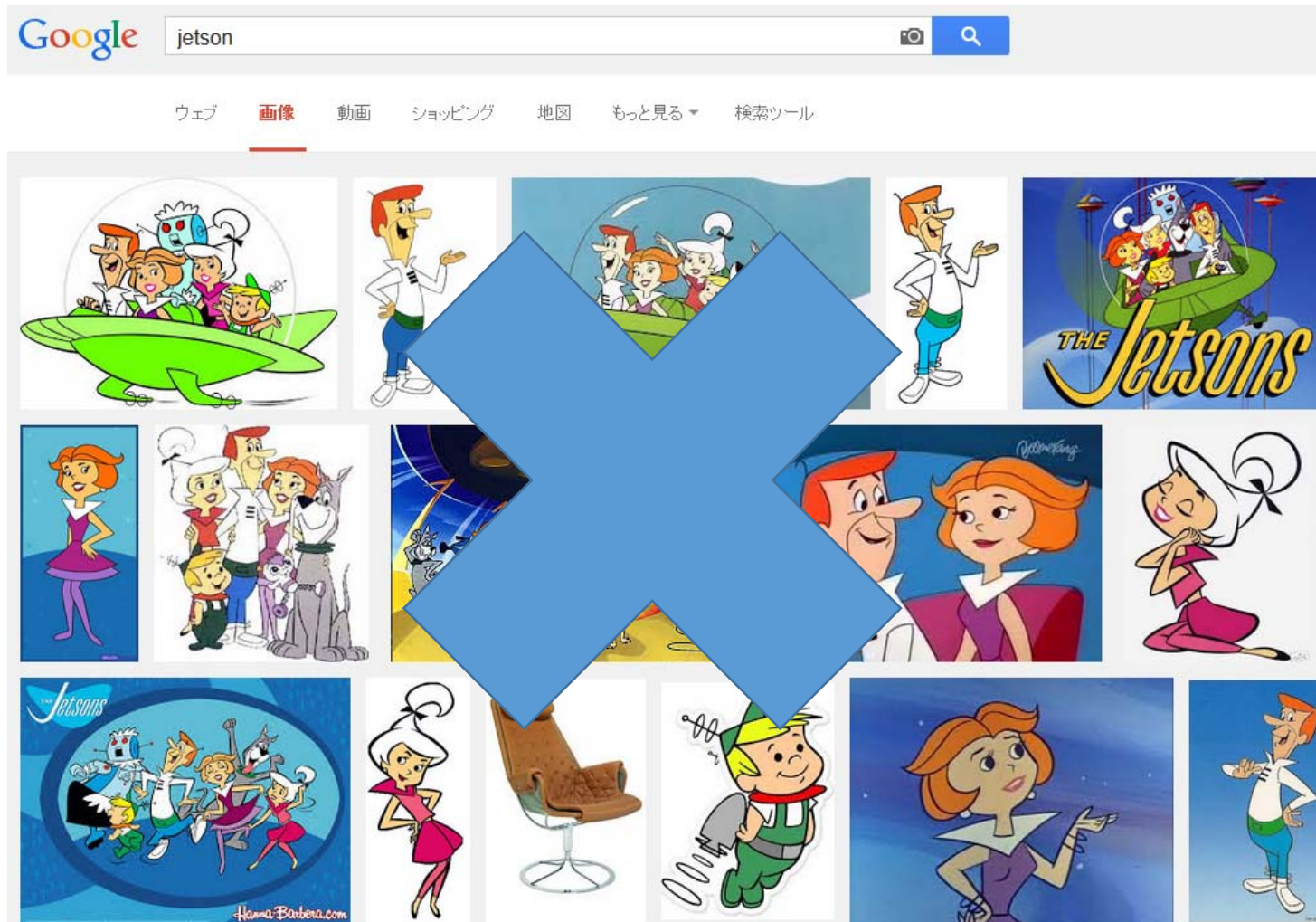
@tomoaki_teshima
CV勉強会@関東

2015. 1. 31

今日の要約

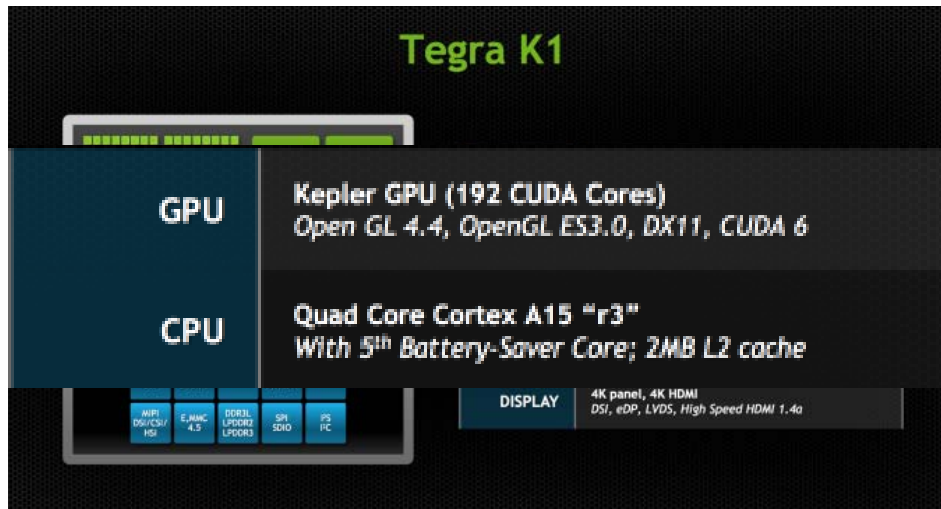
- Jetson、NVIDIAの紹介
 - Jetson上でOpenCVをビルド、走らせてみる
 - OpenCVやARMに関する小ネタ紹介
-
- 注：私はNVIDIAやARMの回し者ではありません

Jetsonとは



Jetson とは

- 組み込み用のNVIDIA製GPU、Tegraの開発ボード
- 4コアのCPU(ARM Cortex-A15)と、
- 192コアのGPU(Kepler)が搭載されている
- 192\$で販売されている



<http://www.bdti.com/InsideDSP/2014/04/29/NVIDIA>

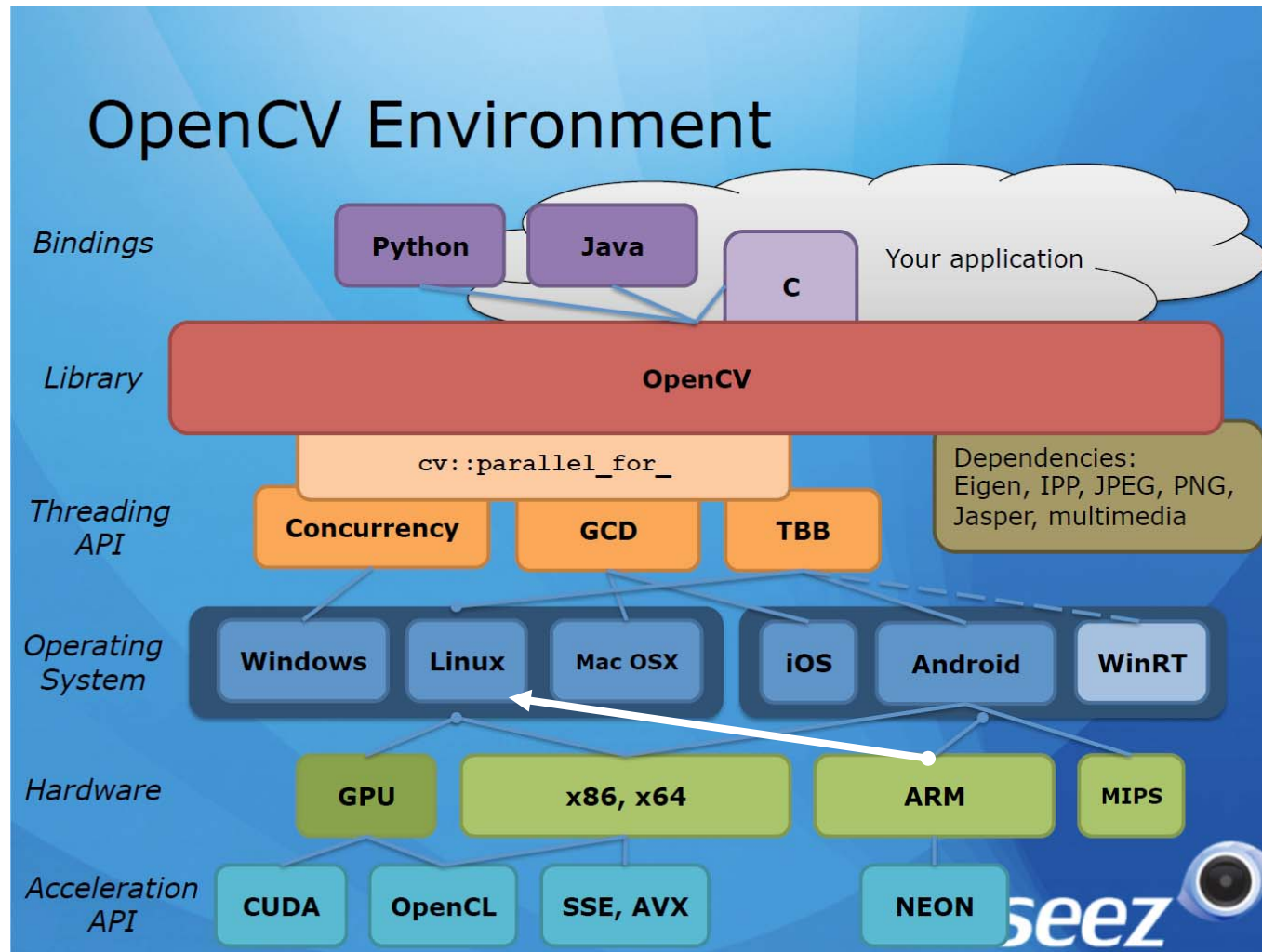
Keplerが乗っている

CUDAがそのまま走る！

OpenCVとは

- 言わずと知れたコンピュータビジョンライブラリ
- そろそろ3.0のmajor release を控えている
- (~~出る出る詐欺~~)
- 様々なプラットフォームがサポートされている

OpenCVのプラットフォーム



<http://www.slideshare.net/EugeneKhvedchenya/opencv-30-latest-news-and-the-roadmap>

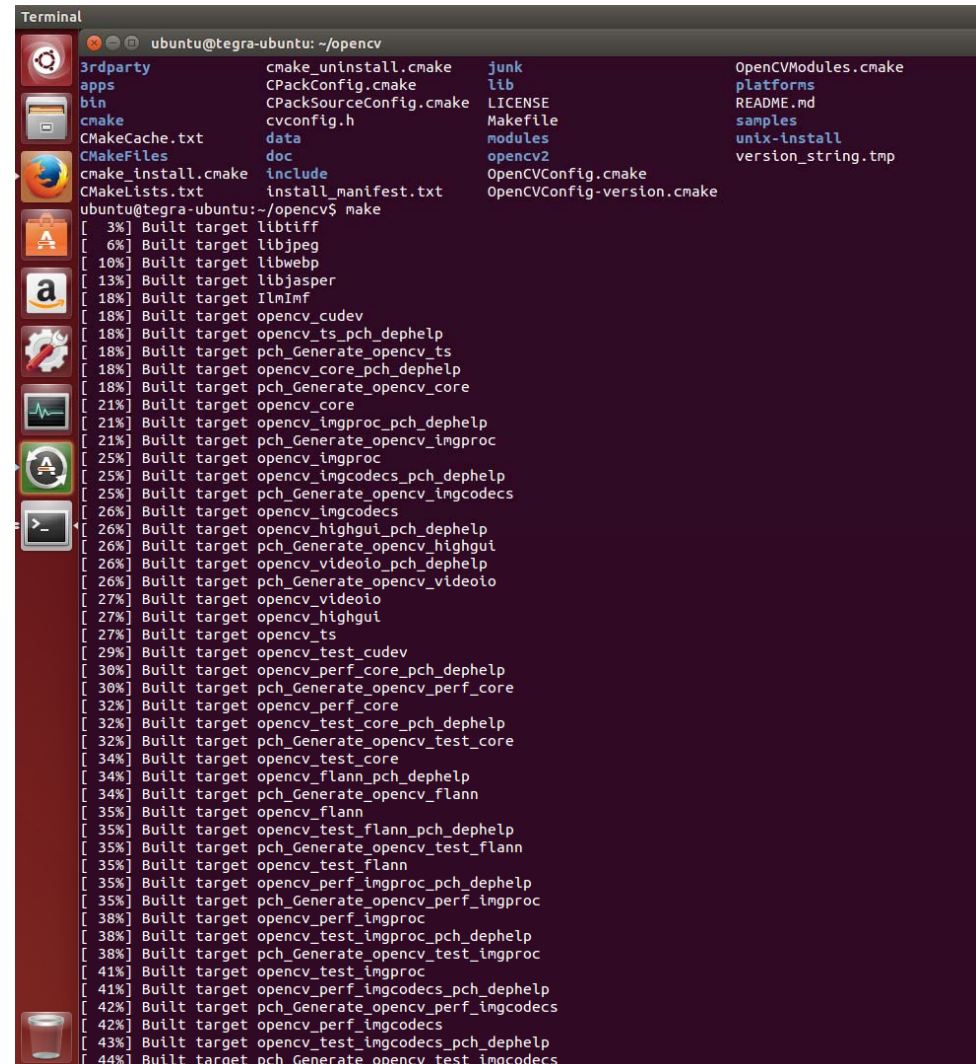
Jetson上でLinuxが走る

すごい

*)ARM上でLinuxが走るボードはJetsonよりも前から沢山ありました

OpenCVのビルド

- NVIDIAからTegra用のパッケージが出ている
- パッケージを使う代わりにコードからビルド
- ビルド自体もJetson上で行う



The image shows a terminal window on a Tegra device (ubuntu@tegra-ubuntu) running the 'make' command to build OpenCV. The terminal output displays a progress bar for various targets, including library dependencies like libtiff, libjpeg, libwebp, libjasper, and Imf, as well as core OpenCV modules like opencv_core, opencv_imgproc, opencv_imgcodecs, opencv_highgui, and opencv_videoio. The build is progressing through approximately 44% of the targets.

```
Terminal
ubuntu@tegra-ubuntu: ~/opencv
3rdparty      cmake_uninstall.cmake  junk          OpenCVModules.cmake
apps          CPackConfig.cmake     lib           platforms
bin           CPackSourceConfig.cmake LICENSE       README.md
cmake         cvconfig.h            Makefile     samples
CMakeCache.txt data                  modules      unix-install
CMakeFiles   doc                  opencv2      version_string.tmp
cmake_install.cmake include              OpenCVConfig.cmake
CMakeLists.txt install_manifest.txt OpenCVConfig-version.cmake

ubuntu@tegra-ubuntu:~/opencv$ make
[ 3%] Built target libtiff
[ 6%] Built target libjpeg
[ 10%] Built target libwebp
[ 13%] Built target libjasper
[ 18%] Built target Imf
[ 18%] Built target opencv_cudev
[ 18%] Built target opencv_ts_pch_dephelp
[ 18%] Built target pch_Generate_opencv_ts
[ 18%] Built target opencv_core_pch_dephelp
[ 18%] Built target pch_Generate_opencv_core
[ 21%] Built target opencv_core
[ 21%] Built target opencv_imgproc_pch_dephelp
[ 21%] Built target pch_Generate_opencv_imgproc
[ 25%] Built target opencv_imgproc
[ 25%] Built target opencv_imgcodecs_pch_dephelp
[ 25%] Built target pch_Generate_opencv_imgcodecs
[ 26%] Built target opencv_imgcodecs
[ 26%] Built target opencv_highgui_pch_dephelp
[ 26%] Built target pch_Generate_opencv_highgui
[ 26%] Built target opencv_videoio_pch_dephelp
[ 26%] Built target pch_Generate_opencv_videoio
[ 27%] Built target opencv_videoio
[ 27%] Built target opencv_highgui
[ 27%] Built target opencv_ts
[ 29%] Built target opencv_test_cudev
[ 30%] Built target opencv_perf_core_pch_dephelp
[ 30%] Built target pch_Generate_opencv_perf_core
[ 32%] Built target opencv_perf_core
[ 32%] Built target opencv_test_core_pch_dephelp
[ 32%] Built target pch_Generate_opencv_test_core
[ 34%] Built target opencv_test_core
[ 34%] Built target opencv_flann_pch_dephelp
[ 34%] Built target pch_Generate_opencv_flann
[ 35%] Built target opencv_flann
[ 35%] Built target opencv_test_flann_pch_dephelp
[ 35%] Built target pch_Generate_opencv_test_flann
[ 35%] Built target opencv_test_flann
[ 35%] Built target opencv_perf_imgproc_pch_dephelp
[ 35%] Built target pch_Generate_opencv_perf_imgproc
[ 38%] Built target opencv_perf_imgproc
[ 38%] Built target opencv_test_imgproc_pch_dephelp
[ 38%] Built target pch_Generate_opencv_test_imgproc
[ 41%] Built target opencv_test_imgproc
[ 41%] Built target opencv_perf_imgcodecs_pch_dephelp
[ 42%] Built target pch_Generate_opencv_perf_imgcodecs
[ 42%] Built target opencv_perf_imgcodecs
[ 43%] Built target opencv_test_imgcodecs_pch_dephelp
[ 44%] Built target pch_Generate_opencv_test_imgcodecs
```

過去の努力

- シングルコアのノートPCで乗車実験をしたM1の冬
- 4kgぐらいする重量級のノートPCを持って、高速道路を走った
- 涙ぐましい最適化をして、なんとかQVGA 4fps を達成してた



ノートPC



webcam

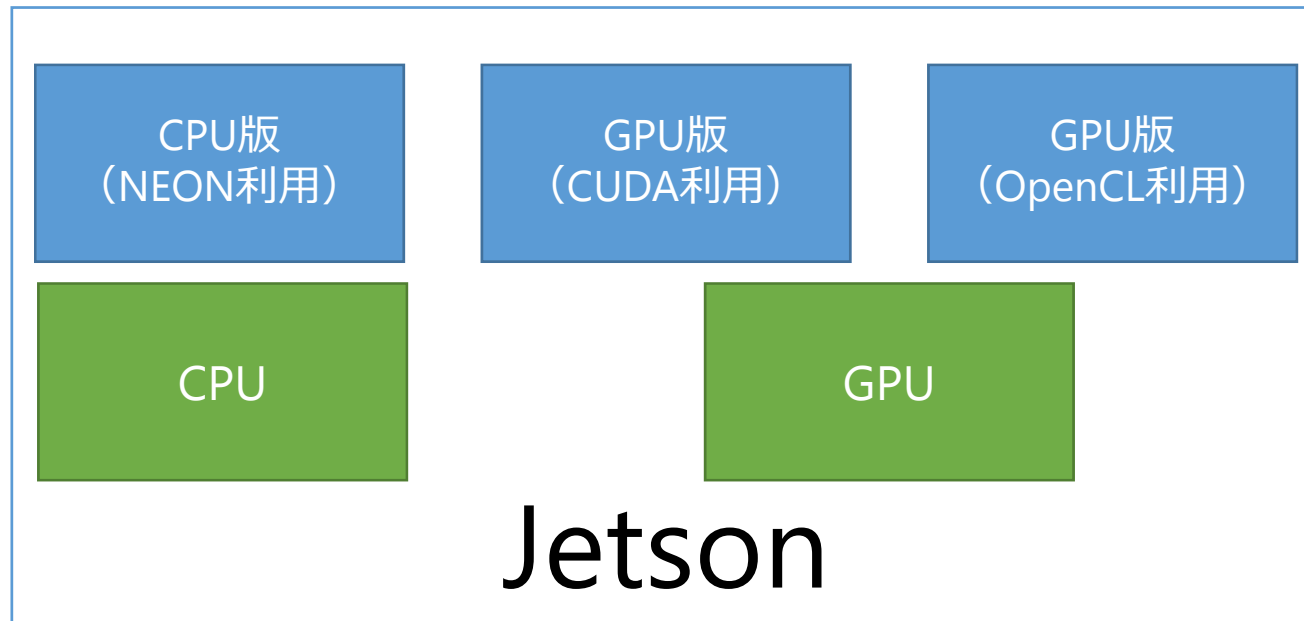
10年経ってみて

あれ？これJetsonでできたんじゃない？

パフォーマンス測定

- NEON最適化(CPU)
- CUDA 6.5
- OpenCL 1.2
- を有効にしてOpenCV-3.0-betaをビルド
- 簡単な処理をしてみても所要時間を計測
 - 残念ながらOpenCLのコンフィグに失敗したようで、そこはまた今度に。
- 参考までに、Corei7 + GT650MのMBPとも比較

パフォーマンス測定

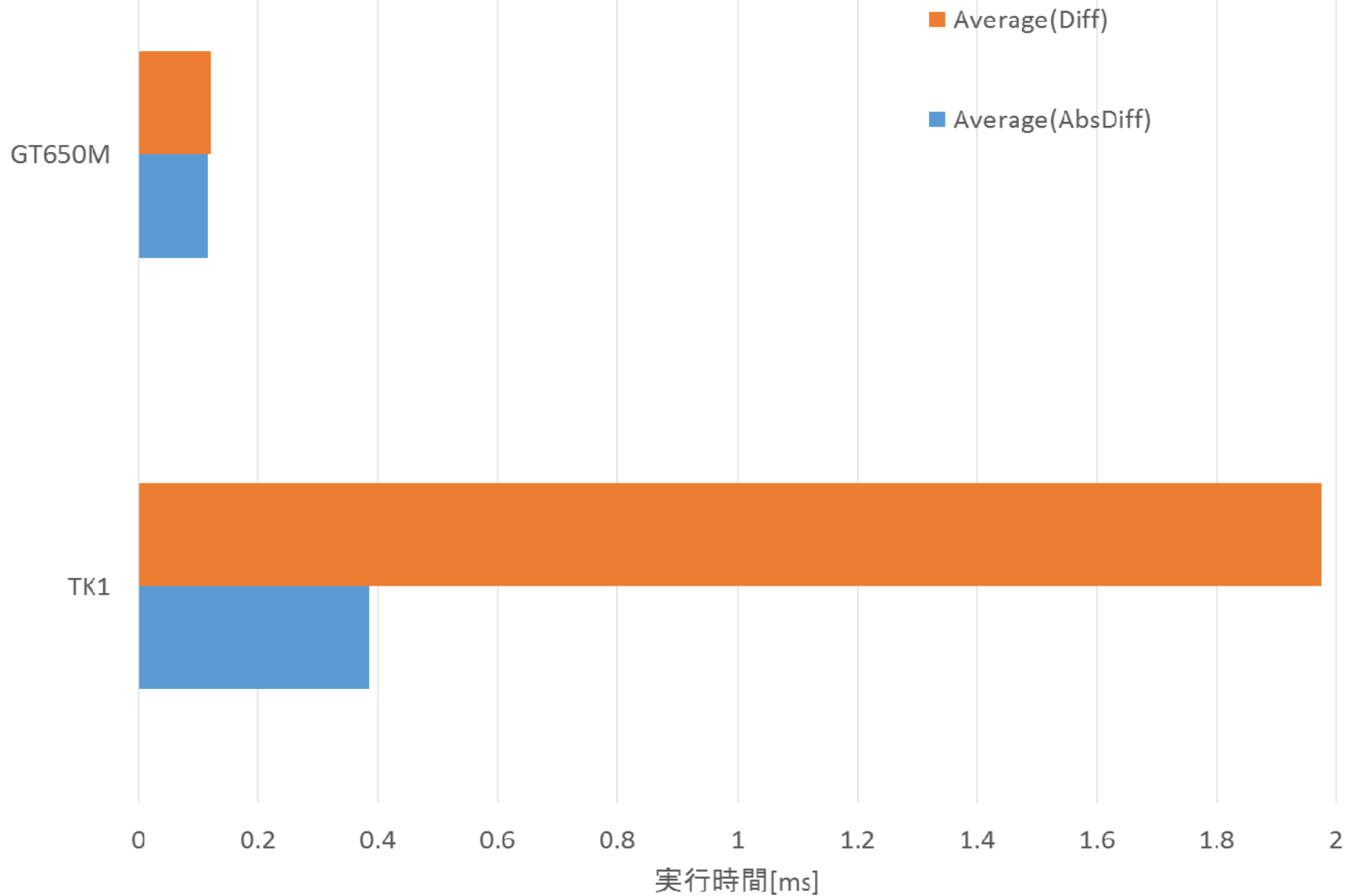


測定方法

- 50回処理した場合と10回処理した場合の差分から、所要時間を測定



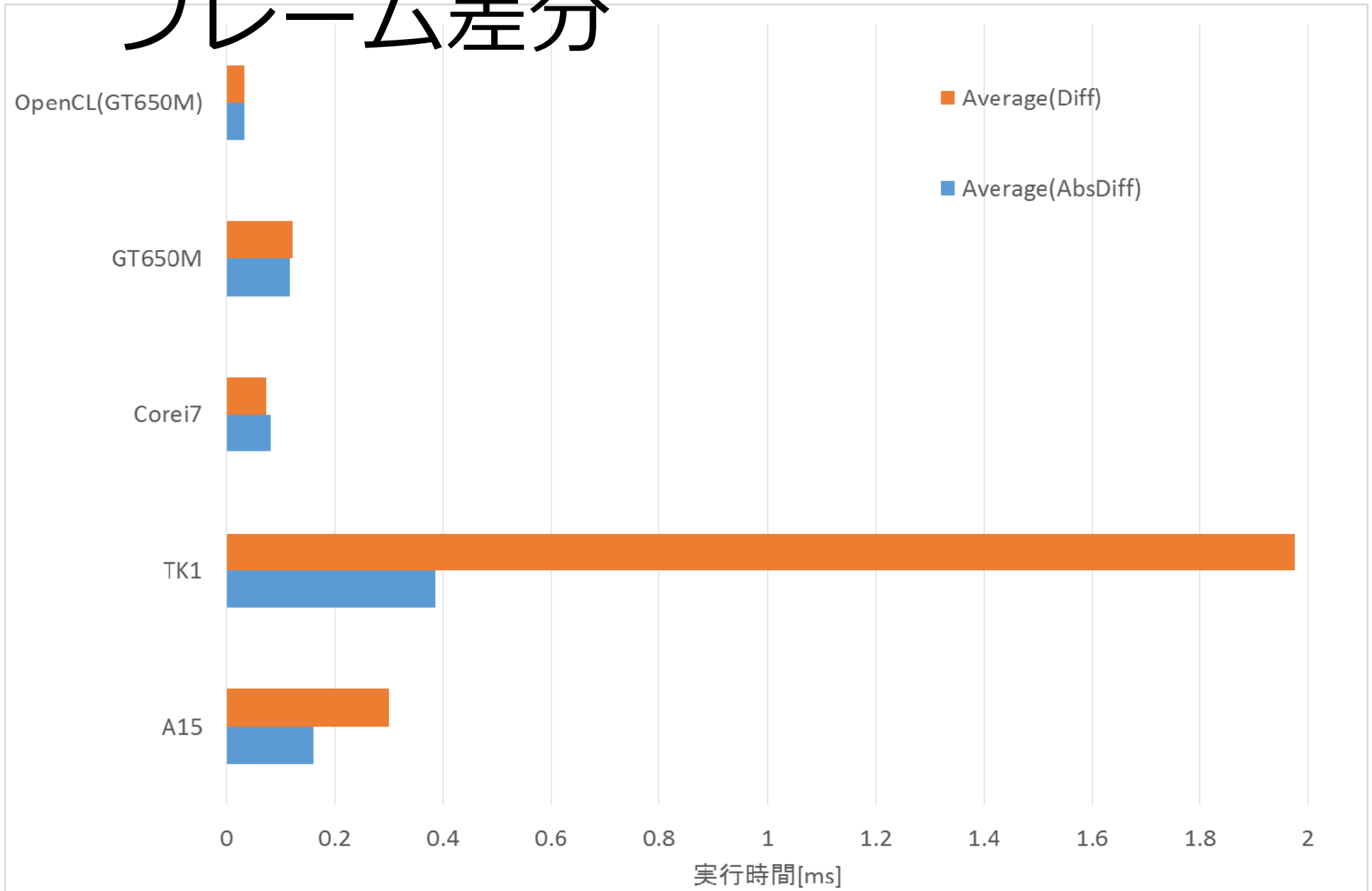
フレーム差分



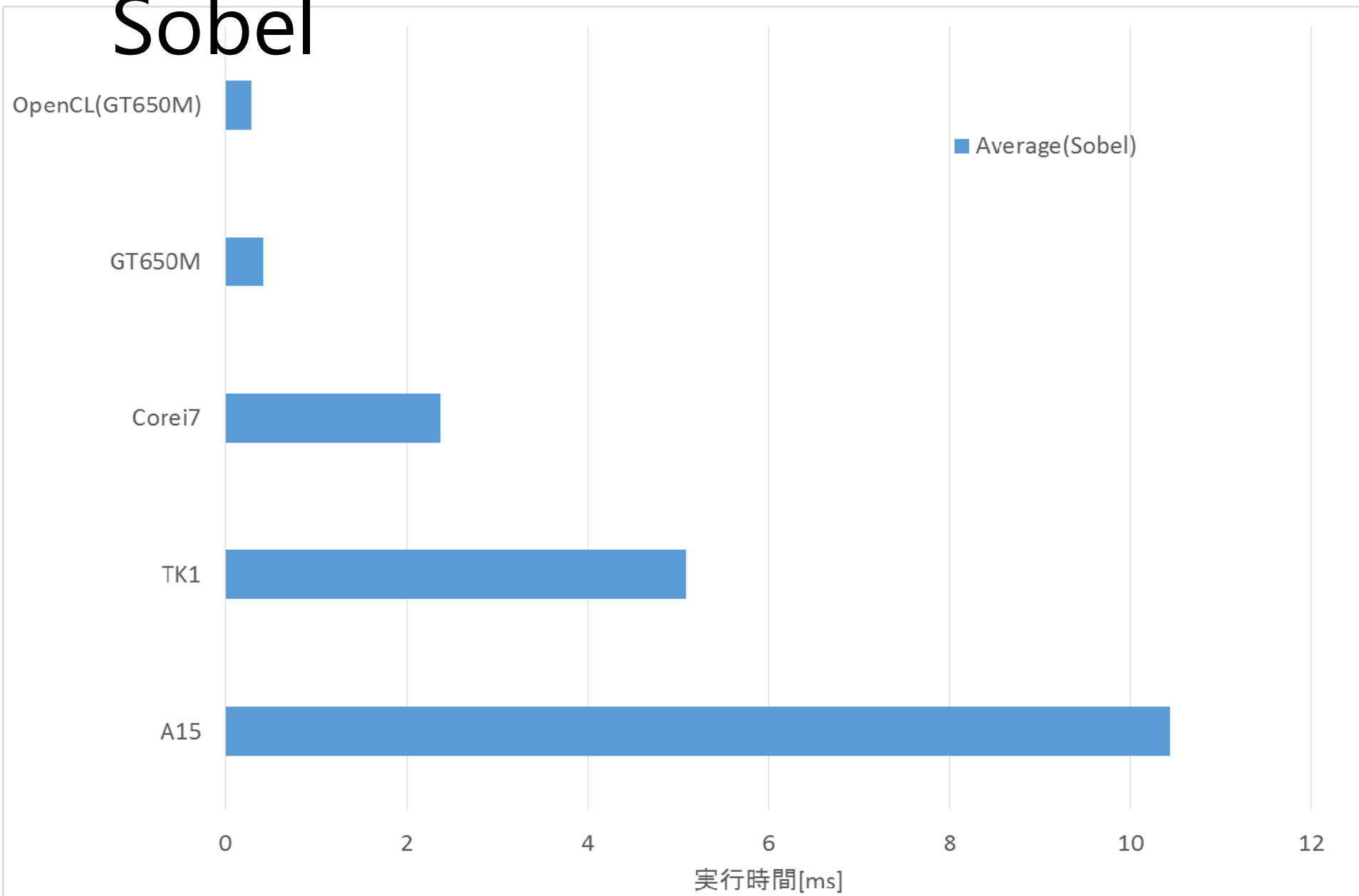
フレーム差分の結果

- さすがに650Mには及ばない
- しかし、対値段効果、対消費電力効果はTK1の方が上（らしい）
- absdiff 関数とただのsubtract でここまで差が出る原因は謎

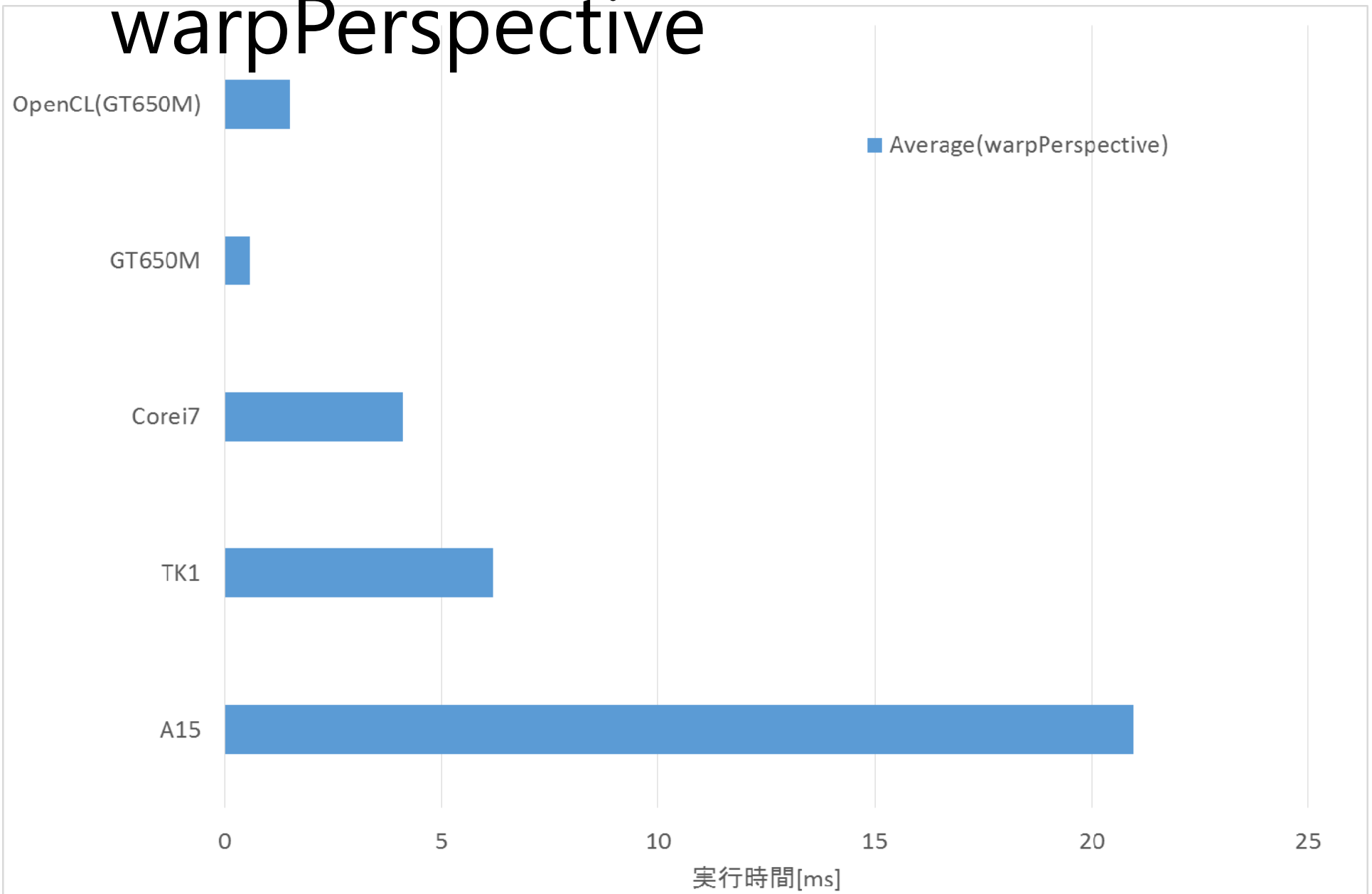
フレーム差分



Sobel



warpPerspective



比較結果

- 192\$でこのパフォーマンス！
- 12V1A ぐらいの電力使用量（らしい）
- 絶対的な性能はデスクトップマシンには今ひとつ及ばない。
- もともと車載向けのSoC

小ネタ (書き方)

CPU (cv::Mat)

```
using namespace cv;
Mat homography;

Mat input = imread(filename);

Mat result;
warpPerspective(input, result,
                homography, input.size());
```

CUDA (cv::cuda::gpuMat)

```
using namespace cv;
Mat homography;

Mat input = imread(filename);
cvtColor(input.clone(), input, COLOR_RGB2RGBA);

cuda::GpuMat gpuInput = cuda::GpuMat(input);
cuda::GpuMat gpuResult;
cuda::warpPerspective(gpuInput, gpuResult,
                    homography, gpuInput.size());
```

OpenCL (cv::UMat)

```
using namespace cv;
Mat homography;

Mat input = imread(filename);

UMat umatInput = input.getUMat(ACCESS_READ);
UMat umatResult;
warpPerspective(umatInput, umatResult,
                homography, umatInput.size());
```

このあたりはdandelion1124先生という方が詳しいです

小ネタ (ARM)

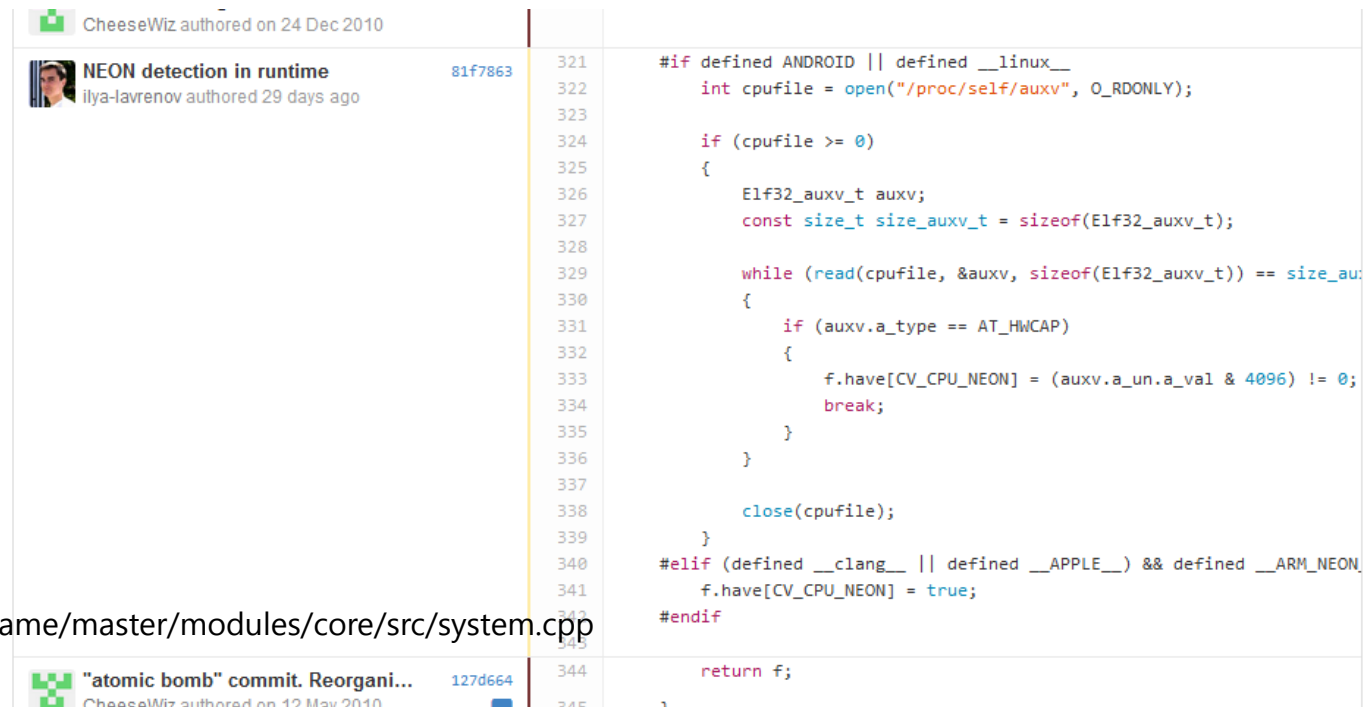
- ARM社は設計をライセンス販売する
- Intelと違い、チップを売っているのではない
- サードパーティー性のARMの互換チップもある
 - Apple (Apple A5-A8)
 - Qualcomm (Snapdragon)
 - Marvell (Sheeva)

小ネタ (NEON)

- ARMv7シリーズから実装されたSIMD命令
- NEON の実行時判定が実装されてなかったの
で、実装してみた
- AndroidかLinuxならば
 - /proc/cpuinfo
 - /proc/self/auxv
 - あたりを調べればわかる
- getauxval を使って調査コードを書いてみた

小ネタ (NEON)

- よしプルリクだ！
- betaには実装されていないが、masterで年末にコミット済みorz....



The screenshot shows a GitHub commit diff for the file `system.cpp`. The commit is titled "NEON detection in runtime" and was authored by `ilya-lavrenov` 29 days ago. The diff shows changes to the file, with line numbers 321 through 345 visible. The code includes a function that checks for NEON support on Android and Linux. It opens `/proc/self/auxv` and reads the `AT_HWCAP` field to determine if NEON is available. The code is as follows:

```
321 #if defined ANDROID || defined __linux__
322     int cpufile = open("/proc/self/auxv", O_RDONLY);
323
324     if (cpufile >= 0)
325     {
326         Elf32_auxv_t auxv;
327         const size_t size_auxv_t = sizeof(Elf32_auxv_t);
328
329         while (read(cpufile, &auxv, sizeof(Elf32_auxv_t)) == size_auxv_t)
330         {
331             if (auxv.a_type == AT_HWCAP)
332             {
333                 f.have[CV_CPU_NEON] = (auxv.a_un.a_val & 4096) != 0;
334                 break;
335             }
336         }
337
338         close(cpufile);
339     }
340 #elif (defined __clang__ || defined __APPLE__) && defined __ARM_NEON
341     f.have[CV_CPU_NEON] = true;
342 #endif
343
344     return f;
345 }
```

<https://github.com/Itseez/opencv/blame/master/modules/core/src/system.cpp>

小ネタ (ARM)

メーカー	機種名	SoC	購入年	CPU	アーキテクチャ	NEON
HTC	HT-03A	MSM7201A	2010	ARM11	ARMv6	NO
Samsung	SC-01C	Exynos3 S5PC110	2010	Cortex-A8	ARMv7-A	YES
Samsung	SC-02C	Exynos4 4210	2011	Cortex-A9	ARMv7-A	YES
LG	Nexus5	Snapdragon 800	2013	Krait 400	ARMv7-A	YES
SONY	SO-02F	Snapdragon 800	2012	Krait 400	ARMv7-A	YES
HTC	Nexus7	Tegra3 30L	2012	Cortex-A9	ARMv7-A	YES
nVidia	Jetson	Tegra K1	2014	Cortex-A15	ARMv7-A	YES
QNAP	TS-409	Feroceon	2010	Sheeva	ARMv5TE	NO

<http://tessy.org/wiki/index.php?ARM>

小ネタ (cuda::threshold)

- マルチチャンネル実装
- しかし、テストを書いている最中
- どうしてこうなった。。

Allow threshold function to handle multi-channel image #3647

[Open](#) tomoaki0705 wants to merge 2 commits into `itseez:master` from `tomoaki0705:master`

Conversation 5 Commits 2 Files changed 1



tomoaki0705 commented 2 days ago

- remove the run time check of channel number of input image
- treat the input image as single channel

Allow threshold function to handle multi-channel image [a4fba6](#)

jet47 was assigned by [ilya-lavrenov](#) 2 days ago



tomoaki0705 commented 2 days ago

The threshold function of CPU handles the multi-channel image as input. This PR will allow CUDA version of threshold to treat multi-channel image, too.

jet47 commented on an outdated diff 2 days ago [Show outdated diff](#)

reshape to single channel correctly [4768b03](#)

jet47 commented on the diff 2 days ago

最適化と実装と研究

- 学生時代
 - 最適化したい！高速化したい！
 - ただの最適化は「研究」としては評価されない



研究で最適化して良い状況

- アルゴリズムで高速化する場合
- 自分のプログラムを利用する状況が非常に限定されていて、かつ高速化が必須の場合
- 息を吸う様にSIMDが書ける人
- 息を吸う様にOpenCLが書ける人
- 息を吸う様に680倍に高速化する人
- 「時間にとってプログラミングの高速化がしたい！」という人はそれに手を出すのは止めておいた方が良い

車輪の再発明

- 研究においては避けた方が良い
- なるべくならOpenCVの実装を使いまわした方がtotalで近道
- 理解するためにあえて実装する or コードを読むはOK

まとめ

- お祝いにJetsonをもらいました
- OpenCVでプログラムを走らせてみた
- CUDAも走るよ！

自己紹介

- 都内で勤務するCVエンジニア
- 勉強会に第1回から参加して
- OpenCVのソースコードを解説したWiki
- <http://tessy.org/wiki>
- https://twitter.com/tomoaki_teshima
- <http://www.slideshare.net/tomoaki0705>

参考文献

- <http://www.slideshare.net/FukushimaNorishige/ssii-33751285>
Fukushima Norishige, マルチコアを用いた画像処理
- <http://d.hatena.ne.jp/wosugi/20131208>
杉本 憲治郎, ハミング距離の計算はホントに速いのか
- <http://www.slideshare.net/YasuhiroYoshimura/gpgpu-dandelion1124-201301130>
Yasuhiro Yoshimura, 【関東GPGPU勉強会#3】 OpenCVの新機能 UMatを先取りしよう
- <http://ja.wikipedia.org/wiki/ARM%E3%82%A2%E3%83%BC%E3%82%AD%E3%83%86%E3%82%AF%E3%83%81%E3%83%A3>
ARMアーキテクチャ
- <http://www.slideshare.net/EugeneKhvedchenya/opencv-30-latest-news-and-the-roadmap>
Eugene Khvedchenya, OpenCV 3.0 - Latest news and the Roadmap
- <http://www.bdti.com/InsideDSP/2014/04/29/NVIDIA>
- <http://man7.org/linux/man-pages/man3/getauxval.3.html>
- <https://github.com/Itseez/opencv>
- http://elinux.org/Jetson/Computer_Vision_Performance#Power_draw_during_computer_vision_tasks