Intel Do-It-Yourself Challenge OpenCV

Nicolas Vailliet
www.Intel-Software-Academic-Program.com
paul.guermonprez@intel.com
Intel Software
2014-02-01

OpenCV ?

Open Computer Vision

OpenCV is a well known computer vision library, written in C++. Intel Galileo support this library.
If you connect a camera or upload pictures on your board, you can analyze or modify images.

Fast development cycle

You can code and compile on your workstation,
run your software at full speed, as often as you want
then compile for Galileo and validate on the board.

OpenCV BW Sample

Procedure

Toolchain

We assume here you just did the procedure about building the cross compile toolchain and you are able to compile with ${CC} or ${CXX}.

Source

Create a file named hellocv.cpp and copy/paste the code from the next slide. Save the file.

This piece of code comes from OpenCV Tutorials.

We encourage you to visit and read quickly all tutorials to have the best panorama view of what OpenCV can do for you.

http://docs.opencv.org/doc/tutorials/tutorials.html

Source Code

```
#include <cv.h>
#include <highgui.h>

using namespace cv;

int main( int argc, char** argv )
{
        char* imageName = argv[1];
        Mat image; image = imread( imageName, 1 );

        if( argc != 2 || !image.data ) { printf( " No image data \n " ); return -1; }

        Mat gray_image;
        cvtColor( image, gray_image, CV_BGR2GRAY );
        imwrite( “reslut.jpg", gray_image );
        return 0;
}
```

Compiling and linking

Compile for Galileo

Compile and link your program with the following command:

```
${CXX}  hellocv.cpp `pkg-config opencv  --cflags --libs` -o hellocv
```

Can't execute the binary locally

You cannot run your program on your workstation as it is not compatible with your OS : libraries are linked as if they were on your board, not the way they are in your OS.

Try : ldd hellocv

You see some libraries missing …

.

Transfer, execute

Transfer and connect

scp hellocv root@192.168.xx.xx:~
ssh root@192.168.xx.xx
chmod +x hellocv

Download sample image, execute, get result image

wget http://docs.opencv.org/_images/Hough_Lines_Tutorial_Original_Image.jpg –O pic1.jpg
./hellocv pic1.jpg
logout *# going back to your workstation*
scp root@192.168.xx.xx:~/result.jpg .

open the result.jpg file to see the results

Input, Output

intel
Software

Note about graphical components

Forward GUI

Some OpenCV samples are not command line only as they use the highgui library. You may want to forward the graphical calls to your workstation using the –X flag for ssh.

Disable GUI

Another solution is to modify the code to remove all highgui calls and make the software headless.

OpenCV Sobel Sample

Processing Sobel Derivatives

```
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include <stdlib.h>
#include <time.h>
#include <stdio.h>

using namespace cv;

int main( int argc, char** argv ) {
Mat src, src_gray; Mat grad;
int scale = 1; int delta = 0;
 int ddepth = CV_16S; int c;
src = imread( argv[1] );
if( !src.data ) { return -1; }

clock_t tStart = clock();

GaussianBlur( src, src, Size(3,3), 0, 0, BORDER_DEFAULT );
cvtColor( src, src_gray, CV_RGB2GRAY );
Mat grad_x, grad_y;
Mat abs_grad_x, abs_grad_y;
Sobel( src_gray, grad_x, ddepth, 1, 0, 3, scale, delta, BORDER_DEFAULT );
convertScaleAbs( grad_x, abs_grad_x );
Sobel( src_gray, grad_y, ddepth, 0, 1, 3, scale, delta, BORDER_DEFAULT );
convertScaleAbs( grad_y, abs_grad_y );
addWeighted( abs_grad_x, 0.5, abs_grad_y, 0.5, 0, grad );

printf("Time taken: %.3fs\n", (double)(clock()-tStart)/CLOCKS_PER_SEC);
imwrite("result.jpg",grad);
return 0;
}
```

Sobel derivatives is composed by :

- A Gaussian blur
- A grayscale conversion
- Computer derivatives along X and Y
- Adding the two components together

Input, Output, Size, Time

512 x 512 px

0.57 secondes

intel Software

Input, Output, Size, Time

1920 x 1200 px

5.18 secondes

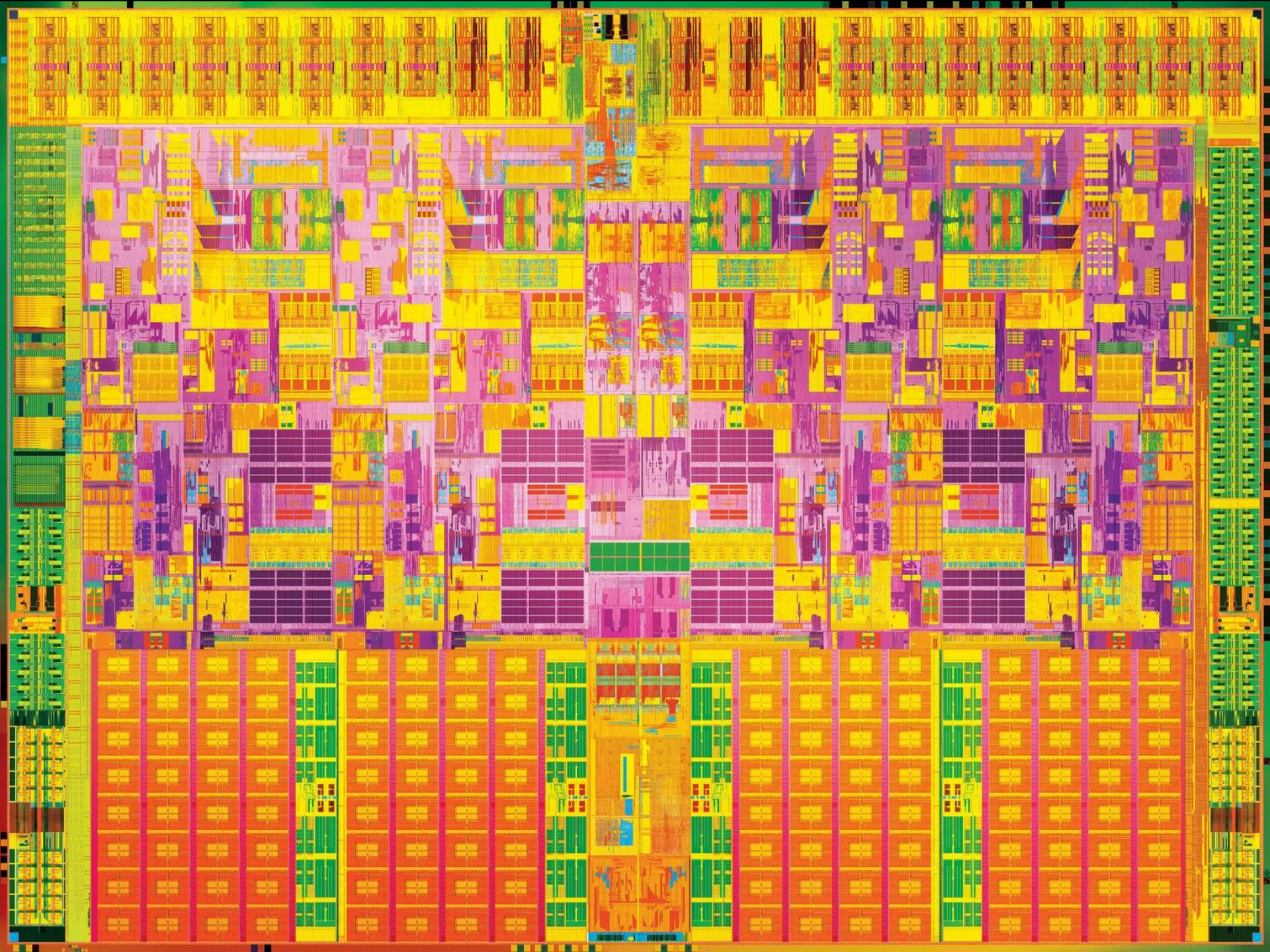
intel Software

Input, Output, Size, Time

300 x 200 px

0.12 secondes

intel Software



License Creative Commons – By 3.0

You are free:

- **to Share** — to copy, distribute and transmit the work
- **to Remix** — to adapt the work
- to make commercial use of the work

Under the following conditions:

- **Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

With the understanding that:

- **Waiver** — Any of the above conditions can be waived if you get permission from the copyright holder.
- **Public Domain** — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.
- **Other Rights** — In no way are any of the following rights affected by the license:
  - Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;
  - The author's moral rights;
  - Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.
- **Notice** — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

http://creativecommons.org/licenses/by/3.0/