**TITLE OF THE THESIS**

**Treball de fi de grau**

**Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona**

**Universitat Politècnica de Catalunya**

**per**

**Ignasi Bernadas Leal**

**GRAU EN ENGINYERIA DE SISTEMES AUDIOVISUALS**

**Tutor: Josep Ramon Casas**

**Barcelona, Juny 2016**

Abstract

Resum

Introducció

Aquest projecte consisteix en la recerca i l’exploració de les possibilitats que pot oferir un sistema incrustat de NVIDIA, la placa Jetson TK1 i, més enllà d’això, fer-la servir com a plataforma per instal·lar-hi un software avançat d’assistència al conductor.

El programa prové d’un projecte de fi de grau d’una estudiant de la Universitat Politècnica de Catalunya amb col·laboració amb l’empresa Arcol i consisteix en un sistema de visió en 360º per a vehicles de grans dimensions.

La visió panoràmica 360º es genera a partir del cosit entre les imatges provinents de 4 càmeres situades al llarg del vehicle, que el dispositiu incrustat ha de processar a temps real per tal de donar suport visual al conductor de l’automòbil.

El problema principal que es va trobar en aquest projecte és que el dispositiu que suportava el programa no tenia prou capacitat computacional com per donar resposta a temps real a l’algorisme de fusió de les imatges.

Davant d'aquest problema es va plantejar la possibilitat d'emprar algun altre dispositiu més potent per tal de millorar els resultats obtingut. Com que l'algorisme de cosit de imatges consisteix en un seguit d'operacions aritmètiques amb matrius, es va pensar que era una bona idea l'ús de una GPU per paral·lelitzar càlculs i augmentar-ne la velocitat.

A partir d'aquí, i aprofitant que la universitat disposava de la placa de NVIDIA, Jetson TK1, i ningú a la universitat l'havia provat com ha eina de desenvolupament, no m'acaba d'agradar el paràgraf.

Desenvolupament del treball

Per tal d'utilitzar la placa Jetson com entorn de desenvolupament i poder executar-hi el software d'Arcol s'ha instal·lat un sistema operatiu amb kernel de Linux (Ubuntu 14.04 LTS). En principi és el que ve pre-instal·lat al dispositiu, però per tal d'obtenir el Know-how de com restaurar el sistema i instal·lar-hi altres llibreries necessàries, s'ha fet un flash del sistema operatiu amb un pack especial per a desenvolupadors proporcionat per NVIDIA. (Jetson Pack).

Un cop configurada la principal eina de treball, també s'ha cregut oportú tenir un entorn de treball més flexible de cara al desenvolupament d'aplicacions, en el qual es poden donar diferents escenaris.

En el primer, i més simple, el mateix dispositiu incrustat serveix com a plataforma de desenvolupament, és a dir, no es requereix de més maquinari extern per programar, compilar i executar el software desenvolupat.

En el segon, la compilació del codi es dur a terme des d’un PC amb una capacitat de càlcul molt més elevada (compilació creuada), cosa que permet treballar amb projectes molt més grans. Per dur a terme la cross-compilació però, s’han instal·lat un seguit de llibreries necessàries per que els softwares compilats siguin compatibles amb l’arquitectura del dispositiu incrustat.

Host – Host

Un cop configurat l’entorn de desenvolupament, s’ha compilat i instal·lat el software.d'Arcol a la placa Jetson.

Després d’instal·lar el software d’Arcol a la placa Jetson s’ha vist que el funcionament sense cap modificació en el codi ja és molt millor que en el hardware original, la placa Xilinx ETC, que funciona amb un processador ARM de 1Gb de memòria Ram.

Tot i la millora, el funcionament del programa encara no és l’adequat, ja que el vídeo sortint mostra al voltant d’una imatge per segon.

A partir d’aquí, s’ha intentat optimitzar el codi de dues maneres; millorant l’algorisme per aprofitar la CPU i, utilitzar la GPU per paral·lelitzar càlculs en les funcions més lentes.

Per fer-ho s'ha fet un "profiling" per determinar quins eren les funcions on es perdia més temps de càlcul.

Degut a la complexitat de tot el projecte, per fer un "profiling" el més real acurat, s'ha extret el la part del software on es fa el cosit de les imatges i s'ha posat en un programa més simple.

En aquest programa s'ha mesurat els temps

Introducció deep learnig??

Resumen

Dedication: A Dedication page may be included in your thesis just before the Acknowledgments page, but it is not a requirement.

Que és això????

Agraïments

ALBERT GIL!!!!! ALBERT GIL!!!!!!!!

JOSEP RAMON!!!! JOSEP RAMON!!!!

Historial de reversions i registre d'aprovació

|  |  |  |
| --- | --- | --- |
| **Revisió** | **Data** | **Motiu** |
| 0 | 10/05/2016 | Creació del document |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

DOCUMENT DISTRIBUTION LIST

|  |  |
| --- | --- |
| **Nom** | **e-mail** |
| Ignasi Bernadas | ibernadasl.iespf@gmail.com |
| Josep Ramon Casas | josep.ramon.casas@upc.edu |
| Albert Gil | albert.gil@upc.edu |
|  |  |
|  |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Escrit per: | | Revisat i aprovat per: | |
| Data | 10/05/2015 | Data |  |
| Nom | Ignasi Bernadas Leal | Nom | Josep Ramon Casas |
| Posició | Autor del projecte | Posició | Supervisor |

Índex

Abstract 1

Resum 2

Resumen 3

Agraïments 5

Historial de reversions i registre d'aprovació 6

Índex 7

Índex de imatges 9

Índex de taules: 10

1. Introducció 11

1.1. Objectius 11

1.2. Requisits i especificacions 11

1.3. Context 11

1.4. Pla de treball 11

2. Tecnologia emprada: 12

2.1. Dispositius Embedded 12

2.2. Computació per GPU 12

3. Desenvolupament del projecte 13

3.1. Configuració entorn de treball 13

3.1.1. NVIDAI Jetson TK1 13

3.1.2. Host (PC) 13

3.1.3. Entorn de compilació i cross-compilació 13

3.1.3.1. ZLIB 13

3.1.3.2. PNG 13

3.1.3.3. TIFF 13

3.1.3.4. JPEG 13

3.1.3.5. Qt 13

3.1.3.6. CUDA 13

3.1.3.7. OpenCv 13

3.2. Porting del Software d’Arcol a la placa Jetson 13

3.2.1. Compilació en el propi dispositiu 13

3.2.2. Compilació creuada 13

3.2.3. Anàlisi i optimització del codi amb CUDA 13

3.2.3.1. Funcions a optimitzar 13

3.2.3.2. Possibles millores 13

3.3. Deep Learning 13

4. Resultats 14

4.1. Software d’Arcol 14

4.2. Deep Learning 14

5. Pressupost ??????? 15

6. Conclusions 16

Bibliografia: 17

Annexos 18

Glossari 19

Índex de imatges

Each figure in the thesis must be listed in the “List of Figures” and each must be given a page number for its easy location.

Índex de taules:

Each table in the thesis must be listed in the “List of Tables” and each must be given a page number for its easy location.

# Introducció

An Introduction that clearly states the rationale of the thesis that includes:

1. Statement of purpose (objectives).
2. Requirements and specifications.
3. Methods and procedures, citing if this work is a continuation of another project or it uses applications, algorithms, software or hardware previously developed by other authors.
4. Work plan with tasks, milestones and a Gantt diagram.
5. Description of the deviations from the initial plan and incidences that may have occurred.

## Objectius

Aprendre el funcionament del dispositiu NVIDIA Jetson TK1

Instal·lar el software d’arcol a la placa Jetson

Millorar i optimitzar el funcionament del sofware respecte el que tenia en hardware pel que es va desenvolupar.

## Requisits i especificacions

## Context

## Pla de treball

# Tecnologia emprada:

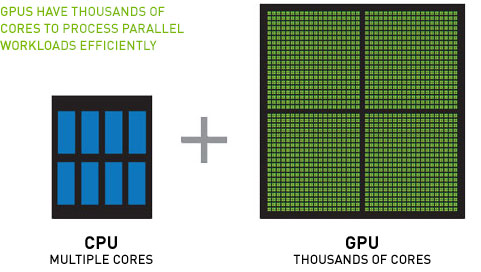
## Dispositius Embedded

Els dispositius incrustats, o “embedded”, son petites computadores dissenyades en general per dur a terme una funció força específica. N’existeixen per una gran varietat d’usos i, és per això que en podem trobar en

A diferència del ordinadors convencionals

## Computació per GPU

Paral·lelització de càlculs

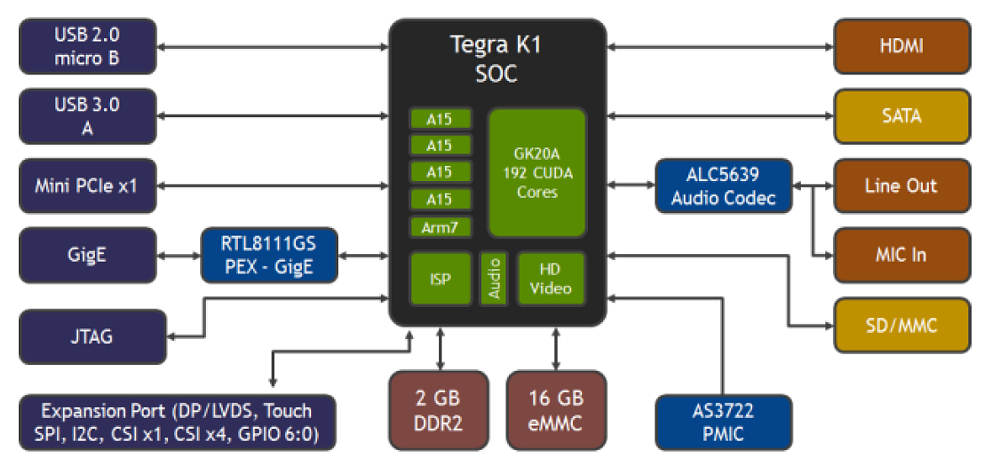


<http://www.nvidia.com/object/gpu-accelerated-computing.html>

# Desenvolupament del projecte

## Configuració entorn de treball

### NVIDAI Jetson TK1



<http://hackerboards.com/nvidia-jetson-tk1-most-advanced-hacker-sbc/>

Jetson Pack

<https://developer.nvidia.com/embedded/downloads>

### Host (PC)

### Entorn de compilació i cross-compilació

#### Eines de compilació

CMake

Scons

#### Llibreries necessàries per compilar arcol

ZLIB

imatge@imatge:~/ibernadas$ git clone https://github.com/madler/zlib.git

imatge@imatge:~/ibernadas/zlib$ cd zlib

imatge@imatge:~/ibernadas/zlib$ git status

imatge@imatge:~/ibernadas/zlib$ git tag

imatge@imatge:~/ibernadas/zlib$ git checkout v1.2.8

imatge@imatge:~/ibernadas/zlib$ ./configure \

CC=arm-linux-gnueabihf-gcc \

--prefix=$ARM\_SDK

imatge@imatge:~/ibernadas/zlib/zlib$ make && make install

PNG

imatge@imatge:~/ibernadas/libpng$ ./configure --host=arm-linux-gnueabihf CC=arm-linux-gnueabihf-gcc CPPFLAGS=-I$ARM\_SDK/include LDFLAGS=-L$ARM\_SDK/lib --prefix=$ARM\_SDK --enable-shared=yes

imatge@imatge:~/ibernadas/libpng$ sudo make && make install

TIFF

imatge@imatge:~/ibernadas$ git clone https://github.com/LuaDist/libtiff.git

imatge@imatge:~/ibernadas$ cd libtiff/

imatge@imatge:~/ibernadas/libtiff$ CC=arm-linux-gnueabihf-gcc \

CXX=arm-linux-gnueabihf-g++ \

CPPFLAGS="-I$ARM\_SDK/include \

CXXFLAGS="-I$ARM\_SDK/include \

CFLAGS="-I$ARM\_SDK/include \

LDFLAGS=-L$ARM\_SDK/lib -mfpu=neon"

./configure \

--prefix=$ARM\_SDK \

--enable-shared=yes \

--host=arm-linux-gnueabihf \

--enable-cxx \

--disable-jbig \

--disable-lzma

imatge@imatge:~/ibernadas/libtiff$ sudo make && make install

JPEG

imatge@imatge:~/ibernadas$ git clone https://github.com/LuaDist/libjpeg.git

imatge@imatge:~/ibernadas/libjpeg$ ./configure \

CC=arm-linux-gnueabihf-gcc \

CPPFLAGS=-I$ARM\_SDK/include \

LDFLAGS=-L$ARM\_SDK/lib \

--prefix=$ARM\_SDK \

--host=arm-linux-gnueabihf \

--enable-shared=yes

imatge@imatge-Alienware-14:~/ibernadas/libjpeg$ sudo make && make install

Qt

Editar el fitxer qt-4.8.2/mkspecs/qws/linux-arm-gnueabi-g++/qmake.conf

#

# qmake configuration for building with arm-none-linux-gnueabi-g++

#

include(../../common/linux.conf)

include(../../common/gcc-base-unix.conf)

include(../../common/g++-unix.conf)

include(../../common/qws.conf)

# modifications to g++.conf

QMAKE\_CC = arm-linux-gnueabihf-gcc

QMAKE\_CXX = arm-linux-gnueabihf-g++

QMAKE\_LINK = arm-linux-gnueabihf-g++

QMAKE\_LINK\_SHLIB = arm-linux-gnueabihf-g++

# modifications to linux.conf

QMAKE\_AR = arm-linux-gnueabihf-ar cqs

QMAKE\_OBJCOPY = arm-linux-gnueabihf-objcopy

QMAKE\_STRIP = arm-linux-gnueabihf-strip

QMAKE\_CFLAGS="${CROSS\_COMPILE\_CFLAGS} -I$(ARM\_SDK)/include -DZYNQ"

QMAKE\_CXXFLAGS="${CROSS\_COMPILE\_CPPFLAGS} -I$(ARM\_SDK)/include -DZYNQ"

QMAKE\_LFLAGS="-L$(ARM\_SDK)/lib"

load(qt\_config)

Embedded (revisar)

imatge@imatge:~/ibernadas$ gunzip qt-everywhere-opensource-src-%VERSION%.tar.gz

imatge@imatge:~/ibernadas$ tar xvf qt-everywhere-opensource-src-%VERSION%.tar

imatge@imatge:~/ibernadas$ cd qt

imatge@imatge:~/ibernadas/qt$ ./configure \

-embedded arm \

-xplatform qws/linux-arm-gnueabi-g++ \

-little-endian -opensource \

-host-little-endian \

-qtlibinfix -UPC-SDK-EMB \

-prefix $ARM\_SDK \

-no-webkit \

-no-phonon \

-no-dbus \

-no-opengl \

-no-pch \

-nomake demos \

-nomake examples \

-system-zlib \

-system-libpng \

-system-libtiff \

-system-libjpeg \

-qt-gfx-transformed \

-qt-gfx-qvfb \

-qt-kbd-qvfb \

-qt-mouse-qvfb -v

imatge@imatge-Alienware-14:~/ibernadas/qt make -j6

imatge@imatge-Alienware-14:~/ibernadas/qt sudo make install

CUDA

OpenCv

imatge@imatge:~/ibernadas$ git clone https://github.com/Itseez/opencv.git

imatge@imatge:~/ibernadas/opencv$ git checkout 2.4.12

imatge@imatge:~/ibernadas/opencv$ mkdir build && cd build

En el fitxer arm-gnueabi.toolchain.cmake canviar la versió del compilador de 4.6 a 4.8 (Explicar)

## Porting del Software d’Arcol a la placa Jetson

### Compilació en el propi dispositiu

### Compilació creuada

### Anàlisi i optimització del codi

#### PROFILING

#### Optimització per CPU

#### Optimització per GPU (CUDA)

#### Possibles millores

## Deep Learning

# Resultats

## Software d’Arcol

## Deep Learning

# Pressupost ???????

Cal posar pressupost?

# Conclusions

This should include your summary, conclusions and recommendations.

Bibliografia:

A thorough reference list as in the following examples: Conference paper [1], journal paper [2], book [3], standard-1 [4], standard-2 [5], online reference [6], patent [7], M.S. thesis [8] and Ph.D. dissertation [9].

1. J. Polastre, R. Szewczyk, D. Culler. "Telos: enabling ultra-low power wireless research". *In Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005*, 25-27 April 2005, Los Angeles, USA. pp. 364-369. doi: 10.1109/IPSN.2005.1440950.
2. V.C. Gungor, B. Lu, G.P. Hancke. "Opportunities and challenges of Wireless Sensor Networks in Smart Grid". *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3557-3564, October 2010. DOI: 10.1109/TIE.2009.2039455.
3. R. Faludi. *Building Wireless Sensor Networks: with ZigBee, XBee, Arduino, and Processing*, 1st ed. Sebastopol, USA: O'Reilly Media, 2010.
4. *Internet Protocol, Version 6 (IPv6) Specification*. IETF RFC 2460, December 1998.
5. *IEEE Standard for Information technology. Telecommunications and information exchange between systems Local and metropolitan area networks. Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Std 802.11-2012.
6. T. Tarun, P. Viswanathan, S. Suman. "Wireless Sensor Network White Paper". *Tetcos Engineering*, 2012. [Online] Available: http://www.tetcos.com/Enhancing\_Throughput\_of\_WSNs.pdf. [Accessed: 23 October 2012].
7. J. P. Wilkinson, “Nonlinear resonant circuit devices,” U.S. Patent 3 624 125, July 16, 1990.
8. M.V. Alvarez Fernández. "Feasibility study and design for Wireless Sensor Networks in a space environment". M.S. thesis, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands, 2011.
9. J. O. Williams, “Narrow-band analyzer,” Ph.D. dissertation, Department of Electrical Engineering, Harvard University, Cambridge, MA, USA,1993.

Annexos

Appendices may be included in your thesis but it is not a requirement.

Glossari

A list of all acronyms and the meaning they stand for.