



**POLITÉCNICA**



**UNIVERSIDAD POLITECNICA DE MADRID  
E.T.S. DE INGENIEROS DE TELECOMUNICACION**

**ACTA DE EXAMEN**

**Asignatura: PROYECTO FIN DE CARRERA**

**TÍTULO DEL PROYECTO:** DESING AND IMPLEMENTATION OF TEXTURE DECOMPRESSION MODULES FOR A GRAPHICS PROCESSING UNIT.

|                     |              |
|---------------------|--------------|
| APELLIDOS Y NOMBRE: | CALIFICACIÓN |
| NOVOA SÚÑER, EMILIO | M.H. 10p     |

**Tutor: D. PABLO ITUERO HERRERO**

Madrid, a 28 de Junio de 2010

**EL VOCAL PRIMERO**

**LA PRESIDENTA**

**EL VOCAL SECRETARIO**

**MIEMBROS DEL TRIBUNAL**

**Presidenta: D<sup>a</sup>. M<sup>a</sup> LUISA LÓPEZ VALLEJO**

**Vocal: D. ÁNGEL FERNÁNDEZ HERRERO**

**Secretario: D. PABLO ITUERO HERRERO**

**TÍTULO:** DESIGN AND IMPLEMENTATION OF TEXTURE  
DECOMPRESSION MODULES FOR A GRAPHICS  
PROCESSING UNIT

**AUTOR:** Emilio Novoa Súaer

**TUTOR:** Pablo Ituero Herrero

**MIEMBROS DEL TRIBUNAL**

- **PRESIDENTE:** M<sup>a</sup> Luisa López Vallejo
- **VOCAL:** Ángel Fernández Herrero
- **SECRETARIO:** Pablo Ituero Herrero
- **SUPLENTE:** Juan Antonio López Martín

**FECHA DE LECTURA:** 28 Junio 2010

**CALIFICACIÓN:** Matriculo de Honor 10p.

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN



PROYECTO FIN DE CARRERA

Design and Implementation of Texture Decompression  
Modules for a Graphics Processing Unit

Emilio Novoa Súañer

2010

## **Resumen:**

En este proyecto se presenta el diseño de descompresores hardware de texturas, de aplicación en la generación de gráficos interactivos en tres dimensiones en una unidad de procesamiento de gráficos (GPU). Tres algoritmos son implementados. En primer lugar se adapta el algoritmo de compresión de imágenes BTC para su uso sobre texturas en tiempo real. BTC es la referencia en cuanto a límite inferior de complejidad para los descompresores. Se implementa asimismo el algoritmo DXT5, cuyo uso está ampliamente extendido en aplicaciones en tiempo real. Se diseña también un descompresor siguiendo el más moderno algoritmo ETC, ideado para arquitecturas empujadas. Los módulos de descompresión son modelados en lenguaje VHDL.

Para su verificación funcional se programan en Matlab un compresor BTC, y descompresores BTC, ETC y DXT5, así como bancos de pruebas en VHDL. Se llevan a cabo simulaciones de los circuitos diseñados en los que son alimentados con datos reales de texturas. El correcto funcionamiento es evaluado a través de la comparación de los resultados de la simulación del hardware con los de los descompresores software, tanto los propios como los de descompresores de terceros.

Tras la verificación funcional, se lleva a cabo la síntesis de los descompresores sobre tecnología FPGA y de células estándar. Esto permite comparar los descompresores en términos de área y retardo. Se demuestra el menor coste del algoritmo ETC respecto a DXT5 para la descompresión de la información de color.

Adicionalmente, se presenta una introducción a la arquitectura segmentada de presentación de gráficos en tres dimensiones por ordenador basada en rasterización de polígonos. El enfoque se dirige hacia la implementación hardware de este tipo de arquitecturas en una GPU. También se analiza la evolución de este tipo de arquitecturas durante los últimos 15 años. Con mayor detalle se trata el tema del uso de texturas: tipos, usos, técnicas de filtrado y los beneficios de la compresión de las mismas para aplicaciones de tiempo real.

## **Palabras clave:**

GPU, texturas, compresión, descompresión, FPGA, células estándar.

## **Summary:**

This thesis presents the design of hardware texture decompressors, which have application in the process of rendering 3D graphics in a graphics processing unit (GPU). Three algorithms are implemented. First, the BTC image compression is adapted for use on textures in real-time graphics. BTC is used as the lower bound on complexity for texture decompressors. The DXT5 algorithm is also implemented, as it is widely used in real-time applications, and has become the practical standard. Another decompressor has been designed following the ETC algorithm, which is more modern and aimed at embedded architectures. All decompressor modules are modeled in VHDL language.

A Matlab BTC compressor, and BTC, ETC and DXT5 compressors have been programmed in Matlab. They are used for functional verification of the VHDL decompressors. Simulations are run in which real texture data is fed to the decompressors. In order to assess the correctness of the designs, the output of the hardware simulation is compared with the results of the software decompressors, both the ones implemented in Matlab and others from third parties.

After their functional verification, the decompressors are synthesized targeting both FPGA and standard cell technologies. This allows to compare the decompressors in terms of area and delay. The lower cost of ETC compared to that of DXT5 for the compression of color data is evidenced.

Additionally, an introduction to a rasterization-based pipelined architecture for rendering 3D graphics is presented. Hints are given on implementation of such an architecture on a GPU. Moreover, the evolution of the GPU architecture during the last 15 years is analyzed. The topic of texturing is revised with more detail: types, uses, filtering techniques and benefits of performing compression.

## **Keywords:**

GPU, textures, compression, decompression, FPGA, standard cell.

# Contents

|   |           |
|---|-----------|
| <b>List of acronims</b>                                 | <b>11</b> |
| <b>1 Introduction</b>                                   | <b>13</b> |
| <b>2 Basics of computer graphics</b>                    | <b>16</b> |
| 2.1 Introduction to rendering . . . . .                 | 17        |
| 2.1.1 Global illumination . . . . .                     | 17        |
| 2.1.2 Local illumination . . . . .                      | 19        |
| 2.2 Graphics APIs . . . . .                             | 21        |
| <b>3 The graphics rendering pipeline</b>                | <b>24</b> |
| 3.1 Application stage . . . . .                         | 25        |
| 3.2 Vertex processing stage . . . . .                   | 28        |
| 3.3 Rasterizer stage . . . . .                          | 31        |
| 3.4 Fragment processing stage . . . . .                 | 33        |
| 3.5 Evolution of the pipeline in GPU hardware . . . . . | 35        |
| <b>4 Texturing</b>                                      | <b>39</b> |
| 4.1 Types of contents for textures . . . . .            | 40        |
| 4.1.1 Color maps . . . . .                              | 40        |
| 4.1.2 Gloss maps . . . . .                              | 40        |
| 4.1.3 Textures for bump mapping . . . . .               | 41        |
| 4.1.4 Shadow maps . . . . .                             | 44        |
| 4.1.5 Cube maps . . . . .                               | 46        |

## CONTENTS

---

|          |   |           |
|----------|---|-----------|
| 4.2      | Texture filtering . . . . .                       | 47        |
| 4.3      | Texture compression . . . . .                     | 51        |
| <b>5</b> | <b>Design of three texture decompressors</b>      | <b>53</b> |
| 5.1      | Presentation of chosen algorithms . . . . .       | 54        |
| 5.1.1    | BTC . . . . .                                     | 54        |
| 5.1.2    | ETC . . . . .                                     | 55        |
| 5.1.3    | DXT5 . . . . .                                    | 55        |
| 5.2      | Objectives, workflow and tools . . . . .          | 56        |
| 5.3      | BTC decompressor . . . . .                        | 58        |
| 5.3.1    | Description of BTC . . . . .                      | 58        |
| 5.3.2    | Compression and compressed image format . . . . . | 60        |
| 5.3.3    | Layout of the BTC decompressor . . . . .          | 60        |
| 5.4      | Design of an ETC decompressor . . . . .           | 61        |
| 5.4.1    | Description of ETC . . . . .                      | 61        |
| 5.4.2    | Compression and compressed image format . . . . . | 62        |
| 5.4.3    | Layout of the ETC decompressor . . . . .          | 62        |
| 5.4.3.1  | Codebook LUT . . . . .                            | 62        |
| 5.4.3.2  | Control block . . . . .                           | 63        |
| 5.4.3.3  | Color calculator . . . . .                        | 63        |
| 5.5      | Design of a DXT5 decompressor . . . . .           | 68        |
| 5.5.1    | Description of DXT5 . . . . .                     | 68        |
| 5.5.1.1  | Color in DXT5 . . . . .                           | 68        |
| 5.5.1.2  | Transparency in DXT5 . . . . .                    | 68        |
| 5.5.2    | Compression and compressed image format . . . . . | 70        |
| 5.5.3    | Layout of the DXT5 decompressor . . . . .         | 70        |
| 5.5.3.1  | DXT5 color calculator . . . . .                   | 71        |
| 5.5.3.2  | DXT5 alpha calculator . . . . .                   | 74        |
| <b>6</b> | <b>Evaluation of the designed decompressors</b>   | <b>79</b> |
| 6.1      | Methodology . . . . .                             | 80        |

## CONTENTS

---

|          |  |           |
|----------|--|-----------|
| 6.2      | Results . . . . .                        | 80        |
| 6.2.1    | BTC decompressor . . . . .               | 81        |
| 6.2.2    | ETC decompressor . . . . .               | 82        |
| 6.2.3    | DXT5 decompressor . . . . .              | 83        |
| 6.2.4    | Comparison and discussion . . . . .      | 84        |
| <b>7</b> | <b>Summary and future work</b>           | <b>87</b> |
| 7.1      | On the learning experience . . . . .     | 88        |
| 7.2      | On the theoretical background . . . . .  | 89        |
| 7.3      | On the decompressors . . . . .           | 89        |
| 7.4      | Future work . . . . .                    | 89        |
| <b>A</b> | <b>Optimization of constant dividers</b> | <b>91</b> |