LARGE DYNAMIC VOXEL SCENES USING SPARSE VOXEL OCTREES

by

NICOLAS FEDOR
URN: 6683787

A dissertation submitted in partial fulfilment of the
requirements for the award of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

May 2025

Department of Computer Science
University of Surrey
Guildford GU2 7XH

Supervised by: Joey Lam

I declare that this dissertation is my own work and that the work of others is acknowledged and indicated by explicit references.

Nicolas Fedor
May 2025

Abstract

Write a summary of the work presented in your dissertation. Introduce the topic and highlight your main contributions and results. The abstract should be comprehensible on its own, and should no contain any references. As far as possible, limit the use of jargon and abbreviations, to make the abstract readable by non-specialists in your area. Do not exceed 300 words.

Acknowledgements

Write any personal words of thanks here. Typically, this space is used to thank your supervisor for their guidance, as well as anyone else who has supported the completion of this dissertation, for example by discussing results and their interpretation or reviewing write ups. It is also usual to acknowledge any financial support received in relation to this work.

Contents

List of Figures

List of Tables

Glossary

A_f	The source message, being a sequence of f source symbols $a_1 a_2 \dots a_f$
a_i	The i^{th} symbol in the source message, where $a_i \in S_m$
B_g	The decoded message, being a sequence of g source symbols $b_1 b_2 \dots b_g$
b_i	The i^{th} symbol in the decoded message, where $b_i \in S_m$
C_h	The transmitted (compressed) message, being a sequence of h Tunstall codewords $c_1 c_2 \dots c_h$
c_i	The i^{th} codeword in the transmitted message, where $c_i \in T_n$
D_h	The received (compressed) message, which for a complete Tunstall code is a sequence of h Tunstall codewords $d_1 d_2 \dots d_h$
d_i	The i^{th} codeword in the received message, where $d_i \in T_n$ for a complete Tunstall code

Abbreviations

BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
BSC	Binary Symmetric Channel
DCT	Discrete Cosine Transform
ECC	Error Correcting Codes
FEC	Forward Error Correction
JPEG	Joint Photographic Experts Group
MPEG	Moving Pictures Experts Group
SER	Symbol Error Rate
SNR	Signal to Noise Ratio

Chapter 1

Introduction

1.1 Dissertation Format

This template is provided to facilitate the process of writing up your dissertation while ensuring its format is consistent with requirements. In using this template, *do not*, under any circumstances, make any changes to the class file provided. In particular, do not attempt to make any changes to the title page, the statement of originality, or the copyright page.

Use of this template is not mandatory; if you choose not to use it, then you must make your final layout resemble this output as closely as possible. In particular, the textual content and layout of the title page, statement of originality, and copyright page must not be changed.

1.2 Using LaTeX

1.2.1 Supported platform

The supported platform for this template is a system with:

- An Ubuntu 10.04 operating system
- The `texlive-full` package installed

This setup is available at least in the Duck/Swan labs, and is supported by FEPS IT. The template itself is kept as simple as possible, to facilitate its use on other systems.

Operation	Speed
Add, Mul, Mul-Add	8
Reciprocal	2
Divide	0.88
Divide Intrinsic	1.6
Recip. Square Root	2
Square Root	1
Logarithm	2
Exponent	1
Sin, Cos Intrinsics	2
Sin, Cos, Tan	Slow

Table 1.1: An example table, showing speed in operations per cycle per multiprocessor

The template also assumes the installation of the GNU make system. To compile the template into a PDF, simply issue the `make` command in the template folder, from a terminal.

1.2.2 Adding figures

The simplest way to add figures is to use the `graphicx` package, as shown in the example that produces Figure 1.1. LaTeX automatically finds a suitable place to lay out the figure. A number of graphic formats are supported – please refer to the package documentation for further details.

1.2.3 Adding tables

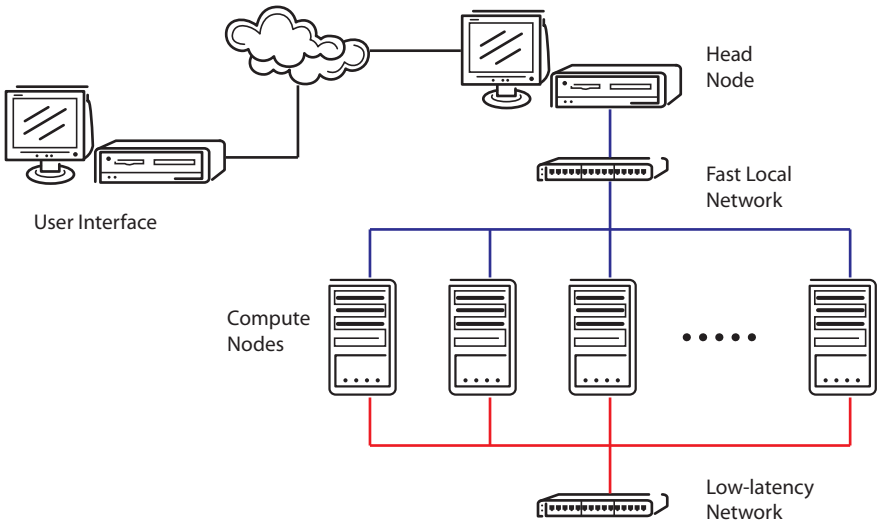
Tables are handled in a similar way to figures, although in this case the text is entered directly in the source file. An example is given in Table 1.1.

1.2.4 Adding equations

One of the strong points of LaTeX is its formatting of mathematical notation. It can easily handle numbered equations, as in the recursive formula:

$$\alpha(\iota, x_2) = \sum_{x_1, d_{\iota-1}} \alpha(\iota - 1, x_1) \Pr\left\{\mathbf{r}_{n(\iota-1)+x_1, n\iota+x_2}, \mathbf{t}_{\iota-1}\right\} \tag{1.1}$$

or math within the main body of text, for example to specify $1 \leq \iota < N$. More complex equations are also handled with ease, as in the following case, where the equation is too long to



(a) Diagrammatic representation

(b) Photo of the Tempest cluster

Figure 1.1: An example figure, with two parts

fit in a single line:

$$\Pr\left\{\mathbf{r}_{0,n\iota+x_2}, \varsigma_{n\iota} = x_2\right\} = \sum_{x_1, d_{\iota-1}} \left[\begin{array}{l} \Pr\left\{\mathbf{r}_{0,n(\iota-1)+x_1}, \varsigma_{n(\iota-1)} = x_1\right\} \\ \times \Pr\left\{\mathbf{r}_{n(\iota-1)+x_1,n\iota+x_2}, \mathbf{t}_{\iota-1}\right\} \end{array} \right] \tag{1.2}$$

1.2.5 Adding code fragments

Code fragments in LaTeX can be added using the `verbatim` environment, as shown in the example below. Remember to set the line spacing as shown, or you'll end up with double-spaced code listings.

```
typedef int v4si __attribute__ ((vector_size (16)));

void ArrayAdd(int *c, const int *a, const int *b, int n)
   {
   const v4si *va = (const v4si *)a;
   const v4si *vb = (const v4si *)b;
   v4si *vc = (v4si *)c;
   int i = 0;
   if(n > 4)
      for(; i < n / 4; i++)
         vc[i] = va[i] + vb[i];
   for(i *= 4; i < n; i++)
      c[i] = a[i] + b[i];
   }
```

1.2.6 Adding references

One of the more convenient facilities provided by LaTeX is the handling of references using BibTeX . This allows you to completely separate the bibliography database from the citations within the text, and more importantly from the way these are formatted. BibTeX support a number of reference types; examples are given in this template for:

- Conference papers (Briffa, Schaathun & Wesemeyer 2010)
- Journal articles (el Gamal, Hemachandra, Shperling & Wei 1987)
- Dissertations (Tunstall 1968)
- Books (Press, Teukolsky, Vetterling & Flannery 1992)
- User or technical manuals (NVI 2009, Henderson 2009)

- Technical Reports (Burrows & Wheeler 1994)
- Other material that cannot be easily classified (Farrell 1992)

Bibliography

Briffa, J. A., Schaathun, H. G. & Wesemeyer, S. (2010), An improved decoding algorithm for the Davey-MacKay construction, *in* 'Proc. IEEE Intern. Conf. on Commun.', Cape Town, South Africa.

Burrows, M. & Wheeler, D. J. (1994), A block-sorting lossless data compression algorithm, Technical report, Digital SRC Research Report.

el Gamal, A. A., Hemachandra, L. A., Shperling, I. & Wei, V. K. (1987), 'Using simulated annealing to design good codes', *IEEE Transactions on Information Theory* **33**(1), 116–123.

Farrell, P. G. (1992), 'Notes on source coding'. University of Manchester.

Henderson, B. (2009), *Netpbm*.
URL: `http://netpbm.sourceforge.net/doc/`

NVI (2009), *NVIDIA CUDA Programming Guide*. Version 2.3.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1992), *Numerical Recipes in C: The Art of Scientific Computing*, second edn, Cambridge Universty Press.

Tunstall, B. P. (1968), Synthesis of Noiseless Compression Codes, PhD thesis, Georgia Institute of Technology.