UNIVERSITY OF FRIBOURG

BACHELOR THESIS

Thesis Title

Author:
Author Name

Supervisor:
Prof. Dr. Philippe Cudré-Mauroux

Co-Supervisor:
Co-supervisor Name

January 01, 1970

eXascale Infolab
Department of Informatics

Boulevard de Pérolles 90 • 1700 Fribourg • Switzerland
phone +41 (26) 300 84 65 • diuf-secr@unifr.ch • www3.unifr.ch/inf

Abstract

Author Name

Thesis Title

Write the thesis abstract here. Should be between half-a-page and one page of text, no newlines.

Keywords: keywords, list, here

Contents

List of Figures

Chapter 1

Chapter Title Here

This introductory file has been edited. Please find the complete version on http://www.latextemplates.com.

Remember to keep your editor's spell checker always on. The preferred spelling is American English; using British English word spelling is accepted only if consistent throughout the thesis.

An invaluable resource when grasping for words is www.thesaurus.com. If a sentence comes more natural in another language, consider using www.deepl.com for translation as the result is typically of higher quality than Google Translate.

1.1 References

The `biblatex` package is used to format the bibliography and inserts references such as this one (**Reference1**). Use `\citet` for textual citations and `\citep` to wrap them in parenthesis (check the source for this text). Multiple references are separated by semicolons (e.g. **Reference2**; **Reference1**) and references with more than three authors only show the first author with *et al.* indicating there are more authors (e.g. **Reference3**). This is done automatically for you.

Scientific references should come *before* the punctuation mark if there is one (such as a comma or period). The same goes for footnotes[1].

1.1.1 A Note on bibtex

The bibtex backend used in the template by default does not correctly handle unicode character encoding (i.e. "international" characters). You may see a warning about this in the compilation log and, if your references contain unicode characters, they may not show up correctly or at all. The solution to this is to use the biber backend instead of the outdated bibtex backend. This is done by finding this command: *backend=bibtex* and changing it to *backend=biber*. You will then need to delete all auxiliary BibTeX files and navigate to the template directory in your terminal (command prompt). Once there, simply type `biber main` and biber will compile your bibliography. You can then compile `main.tex` as normal and your bibliography will be updated. An alternative is to set up your LaTeX editor to compile with biber instead of bibtex, see here for how to do this for various editors.

1.2 Tables

Check the source for an example of the required table style.

[1] Such as this footnote, here down at the bottom of the page.

TABLE 1.1: **Caption.** After a useful title, the caption should describe the figure by itself. A reader should know everything about this table (or figure) without having to look for its description in the text.

	longer one	short	short	short	**bold**
# label 1	~3034	~650	~650	~650	**~18**
# longer label	2	3	3	3	**0**
# label 3	~906k	~436k	~436k	~436k	**~3k**

You can reference tables with `Table~\ref{<label>}` where the label is defined within the table environment, see source of Table 1.1.

1.3 Figures

Same as Tables, check source for example. Keep all figures in the `figures` folder. Strongly prefer vectorial image types (e.g. SVG) embedded into PDFs, over high-resolution lossless (e.g. PNG), over very-high-resolution lossy (e.g. JPG).

FIGURE 1.1: **An electron.** Artist's impression.

Sometimes figures don't always appear where you write them in the source. The placement depends on how much space there is on the page for the figure. Sometimes there is not enough room to fit a figure directly where it should go (in relation to the text) and so LaTeX puts it at the top of the next page. Positioning figures is the job of LaTeX and so you should only worry about making them look good!

Figures should have captions (such as in Figure 1.1). The `\caption` command contains two parts, the first part, inside the square brackets is the title that will appear in the *List of Figures*, and so should be short. The second part in the curly brackets should contain the longer and more descriptive caption text.

The `\decoRule` command is optional and simply puts an aesthetic horizontal line below the image. Avoid if possible, consider wrapping the image in a `\mbox` for borders instead

1.4 Typesetting mathematics

The "Not So Short Introduction to LaTeX" (available on CTAN) should tell you everything you need to know for most cases of typesetting mathematics. If you need more information, a much more thorough mathematical guide is available from the AMS called, "A Short Math Guide to LaTeX" and can be downloaded from: ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf

There are many different LaTeX symbols to remember, luckily you can find the most common symbols in The Comprehensive LaTeX Symbol List.

You can write an equation, which is automatically given an equation number by LaTeX like this:

```
\begin{equation}
E = mc^{2}
\label{eqn:Einstein}
\end{equation}
```

This will produce Einstein's famous energy-matter equivalence equation:

$$E = mc^2 \tag{1.1}$$

All equations you write (which are not in the middle of paragraph text) are automatically given equation numbers by LaTeX. If you don't want a particular equation numbered, use the unnumbered form:

```
\[ a^{2}=4 \]
```

1.5 Sectioning and Subsectioning

You should break your thesis chapters into useful sections and subsections. LaTeX automatically builds a table of Contents by looking at all the `\chapter{}`, `\section{}` and `\subsection{}` commands you write in the source.

1.6 In Closing

For the final submission, generate the pdf then search it for question marks (?). Sometimes latex misses a reference or citation and adds a question mark to fill it. Make sure to fix them all before your submission.

Good luck and have fun!

Guide written by —
Sunil Patel: www.sunilpatel.co.uk
Vel: LaTeXTemplates.com

Chapter 2

Technical information

2.1 DICOM

2.1.1 Origin

The acronym DICOM stands for Digital Imaging and Communications in Medicine. Before the 1980's, images resulting from CT scans and MRIs were only decodable by machine manufacturers, while the medical community needed to export and share them for other tasks. For that reason, the ACR (American College of Radiology) and the NEMA (National Electrical Manufacturers Association) created a committee to build a standard. After two iterations with other names, DICOM was created in 1993. It standardized the representation of medical images and their transmission since it provided a network protocol built on top of TCP/IP.

2.1.2 Data format

DICOM files can be viewed as containers of attributes, also called tags. The values of the pixels themselves are stored under the "Pixel Data" tag. Every single DICOM file usually represents a 2-dimensional image, which will form a 3-dimensional volume when put all together.

Other useful information such as the patient name and ID is directly stored within the DICOM files. This approach aims at linking each image to a specific person and event in order not to mix them up. Each DICOM file can be seen as part of a bigger dataset.

2.1.3 Processing images

When manipulating DICOM files, multiple details must be taken into account.

Order

First of all, the name of the files within datasets is a 6 digit number, from 000000 to the number of images minus one. However, this order doesn't match the real order of the images. In fact, the correct order is given by the "Instance Number" tags contained in the various files. Therefore, converted images must be sorted by instance number.

Data manipulation

CT and MRI machines, as well as monitors, differ from one manufacturer to the other and even from one model to the other. DICOM takes this problematic into account by providing specific tags that allow to display the exact same representation of the data, no matter the hardware used. Otherwise, physicians may struggle to detect anomalies because of color and exposition-related variations. Therefore, before displaying or converting an image to any format (such as png), pixel data must be normalized.

The procedure depends on the tags "Window Width" and "Window Center" (one always come with the other). These are used to represent a range of values corresponding to the pixel values in the data. For instance, a window center of 0 and a window width of 200 imply pixel values between -100 and 100.

If they are missing, a simple conversion is sufficient. The parameters to use to convert the data are given by two tags:

- Bits allocated: the number of bits used to represent a single pixel (value: 1 or a multiple of 8)
- Samples per pixel: the number of channels for each pixel

Examples:

- 1 bit, 1 sample: black and white
- 8 bits, 1 sample: grayscale
- 8 bits, 3 samples: RGB
- 16 bits, 1 sample: grayscale

If they are included in the DICOM header, a linear transformation must be done to convert the stored representation of the pixels to the correct visualizable one. To do this, two step are required:

1. **Apply the Hounsfield correction**
   Hounsfield Units (HU) are used in CT images it is a measure of radio-density, calibrated to distilled water and free air. Provided that the rescale slope and the rescale intercept are included in the DICOM header, the correction is applied thanks to the following formula:

   $$HU = m * P + b \tag{2.1}$$

   where m is the rescale slope, P the pixel value, b the rescale intercept.

2. **Apply a linear transformation**
   The result of the first operation then goes through a linear transformation based on the following conditions:

   $$\text{if } (P \leq c - 0.5 - \frac{w-1}{2}), \text{ then } y = y_{min} \tag{2.2}$$

   $$\text{else if } (P > c - 0.5 + \frac{w-1}{2}), \text{ then } y = y_{max} \tag{2.3}$$

   $$\text{else } y = (\frac{P - (c - 0.5)}{w - 1}) + 0.5) * (y_{max} - y_{min}) + y_{min} \tag{2.4}$$

   where c is the window center, w window width, P the pixel input value, y the pixel output value, y_{min} the minimal value of the output range (usually 0), y_{max} the maximal value of the output range (usually 255). Equations 2.2, 2.3 and 2.4 ensure that the pixel values are correctly distributed within the output range.

2.2 NIfTI

Origin

The Neuroimaging Informatics Technology Initiative (NIfTI) file format is the successor of the ANALYZE file format. The main problem of the latter was that it was lacking information about orientation in space. Therefore, the interpretation of stored data could be problematic and inconsistent. For instance, there was a real confusion to determine the left and right sides of brain images. Hence, the NIfTI file format was defined to overcome this major issue.

Data format

Unlike the ANALYZE format that used two files to store the metadata and the actual data, the NIfTI file format stores them in one single file ".nii" but keeps this split between the real data and the header for compatibility. This has the advantage to facilitate the use of the data and avoid storing the data without the meta-data. The NIfTI format can also be compressed/decompressed on-the-fly using the "deflate" algorithm.

Overview of the header structure

With the goal of preserving the compatibility between the ANALYZE and the NIfTI formats, both headers have the same size of 348 bytes. "Some fields were reused, some were preserved, but ignored, and some were entirely overwritten". Details about the different fields contained in the header can be found here: https://brainder.org/2012/09/23/the-nifti-file-format/

2.3 RAW and MHD

Some datasets use a combination of RAW and MHD files. The latter contain metainformation about their corresponding RAW file(s) which contain the data. In most cases, each MHD file points to a unique RAW file whose name is the same as the MHD file name. A single RAW file can be used to represent three-dimensional data, i.e. the combination of multiple two-dimensional images. Libraries such as SimpleITK in Python allow to manipulate RAW images in an easy way.

2.4 Conversion

Medical data is often represented over 16 bits. However, exporting images to PNG require 8-bit images. To transpose a 16-bit image to an 8-bit one, the following procedure was applied:

1. Find maximum value of the current image $Pixel_{min}$
2. Find minimum value of the current image $Pixel_{max}$
3. Pixel normalization: $Pixel = \frac{(Pixel - Pixel_{min}) * 255.0}{Pixel_{max} - Pixel_{min}}$
4. Modify object type to 8 bits
5. Export to PNG