Institutionen för systemteknik
Department of Electrical Engineering

Examensarbete

Real time textrendering

Examensarbete utfört i Informationskodning
vid Tekniska högskolan vid Linköpings universitet
av

Gustav Adamsson

LiTH-ISY-EX--YY/NNNN--SE

Linköping 2015

Linköpings universitet
TEKNISKA HÖGSKOLAN

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

Real time textrendering

Examensarbete utfört i Informationskodning
vid Tekniska högskolan vid Linköpings universitet
av

Gustav Adamsson

LiTH-ISY-EX--YY/NNNN--SE

Handledare: **Jens Ogniewski**
ISY, Linköpings universitet
Mattias Wingstedt
Visiarc

Examinator: **Ingemar Ragnemalm**
ISY, Linköpings universitet

Linköping, 9 april 2015

Avdelning, Institution
Division, Department

informationskodning
Department of Electrical Engineering
SE-581 83 Linköping

Datum
Date

2015-04-09

Språk
Language

□ Svenska/Swedish
☒ Engelska/English

□ _____

Rapporttyp
Report category

□ Licentiatavhandling
☒ Examensarbete
□ C-uppsats
□ D-uppsats
□ Övrig rapport

□ _____

ISBN
—

ISRN
LiTH-ISY-EX--YY/NNNN--SE

Serietitel och serienummer
Title of series, numbering

ISSN
—

URL för elektronisk version

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-XXXXX

Titel
Title

Realtidstextrendrering
Real time textrendering

Författare
Author

Gustav Adamsson

Sammanfattning
Abstract

Det här som vi har hållit på med är jätteviktigt faktiskt och det vi gjort blev bara sååå bra. Kanske inte helt otippat, men det glass är sååå gott!

Förresten har vi blivit bäst på att skriva rapporter, så nu ska ska vi inte gå in närmare på några detaljer såhär i sammanfattningen.

Nyckelord
Keywords

textrendering, distance field, distance map, distance transform

Abstract

If your thesis is written in English, the primary abstract would go here while the Swedish abstract would be optional.

Acknowledgments

We think alla har varit så himla goa hela den här långa och tuffa tiden i våra liv.

Linköping, Januari 2020
N N och M M

Contents

Notation

Några mängder

Notation	Betydelse
$\mathbb{N}$	Mängden av naturliga tal
$\mathbb{R}$	Mängden av reella tal
$\mathbb{C}$	Mängden av komplexa tal

Förkortningar

Förkortning	Betydelse
ARMA	Auto-regressive moving average
PID	Proportional, integral, differential (regulator)

1

Introduction

Redan de gamla grekerna och rommarna trodde att...

2

Background

This chapter includes relevant background information about the main topics of this report. The purpose of the chapter is to give the reader some basic knowledge and undersstanding about the touched topics to facilitate further reading of the report.

2.1 Computer graphics

Computer graphics is good.

2.2 Basic text rendering

Text rendering is a non trivial subject in computer graphics and has been for a long time. There are many different ways to render text to a screen. A common way is to pre render all the glyphs of a font to an image. Characters of this font can then be rendered by using a part of the image as texture for a polygon. Another way is to use some library to render the text to an image and use it to texture a quad [FreeType, 2014]. Both of the above mentioned methods has some drawback, for example the text can not be scaled up without losing the smooth edges and the images takes a lot of memory if you want to have high resolution text.

When rendering a text it is really important that every glyph is positioned to eachother as specified in the font file. Every glyph in a font has data stored in the font with information. For example the OpenType and the TrueType™ font format has information about of advance, kerning, height, widht etc[Microsoft Corporation, 2015, Apple Inc., 2015].

2.3 Distance fields

Binary images is used frequently in image processing [Ragnemalm, 1993]. A binary image is an image with only one binary value for each pixel. This will limit the image to only contain two values or colors. One example of an image that could be represented as a binary image is a picture of a white letter on a black background. In this case white would be represented as 0 and black as 1 or vice versa.

A distance transform is the transform of an input binary image to an output image called distance field or distance map [Rosenfeld and Pfaltz, 1966]. The values of the distance field pixels represents the closest distance by some distance metric to an arbitrary shape in the input image. The distance metric can differ between different applications but the most commonly used distance metric is the euclidean distance metrics which is also called real distance. If euclidean distance metrics is used when doing the distance transform it is called euclidean distance transform(EDT). Other distance metrics that can be used in distance transforms is the city-block distance metric and the chessboard distance metric. When using city-block distances you are only allowed to travel horizontally or vertically in a grid and with the cost 1. Chessboard distance is an extension of city-block distance where you are also allowed to travel diagonally with the cost of 1.

A distance field can be signed or unsigned. An unsigned distance field only maps the distances either inside the shape or outside the shape while a signed distance field maps the pixels inside the shape with negative distances and the outside of the shape with positive distances or vice versa.

Distance fields has many good advantages compared to other methods for storing shapes like boundary representation. One of the most obvious advantages is that a distance field represents more than just the boundary. For example a signed distance field also represents the environment around the boundary, both the interior and the exterior. This makes it easy to check if a point is inside or outside a shape by checking the value of that point in the distance field. It is also easy to move the boundary of the object by just changing the threshold value determining where the boundary is located. [Jones et al., 2006]

2.4 Beziér curves

A beziér curve is defined in space or the plane as two endpoints and a number of control points which is blended together with one blending functions per point. The number of control points depends on the degree of the beziér curve. The most commonly used beziér curves is the cubic beziér with four points and the quadratic beziér with three points. A cubic beziér is defined as:

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3,$$

where $t \in [0, 1]$, P_0 and P_3 are the endpoints of the beziér and P_1 and P_2 are the

control points of the beziér. The sum of the blending functions is always equal to 1 for any t in the function domain.[Ragnemalm, 2008]

A bezier curve can not represent arbitrary curves, therefore it is important to be able to connect several beziers to a path to be able to describe more complex shapes. When connecting curves, continuity between the curves is an important concept have in mind. Parametric continuity is a measure that will have grave importance of the appearance of the curve. There are three levels of parametric continuity.

C^0 = The curves meet
C^1 = The derivative for the both curves is equal where the curves meet
C^2 = The second derivative for the both curves is equal where the curves meet

Another measure than can be used for continuity is geometric continuity which is almost the same as parametric continuity. The only difference is that geometric continuity only require the derivatives of G^1 and G^2 to be proportional and not equal. Given two cubic beziér curves p defined by p_1, p_2, p_3 and p_4 and q defined by q_1, q_2, q_3 and q_4. Assume that p and q is placed in a way that $p_4 = q_1$. This will trivially fulfill the requirement for C^0 because they share one endpoint and $p(1) = q(0)$. C^1 continuity, $p'(1) = q'(0)$ will be fulfilled if p_3, p_4 and q_1 is located on the same line. C2 continuity $p''(1) = q''(0)$ will be fulfilled if p_3, p_4 and q_1 is located on the same line and $|p_4 - p_3| = |q_2 - q_1|$ meaning that the distance from p_3 to p_4 must be equal to the distance from p_4 to q_2.

An example of an application of beziérs is fonts. Every glyph in a font is stored as a number of straight lines and bezier curves, either cubic or quadratic.[Phinney, 2001] Even though beziérs is used in fonts it is not a trivial problem to draw a beziér to the screen. Beziér curves is hard to draw because at the lowest level graphics hardware can only draw polygons and line segments. This is solved by approximating smooth curves to line segments before drawing. [Shreiner et al., 2009] Another problem with beziér curves is that it time consuming to find the closest point on the beziér curve given an arbitrary point. To solve this problem $(p - q(t))q'(t) = 0$, where $t \in [0, 1]$, p is a point in space and q is a beziér curve, has to be solved[Chen et al., 2007]. This implies that a quintic polynomial needs to be solved to find the closest point on a cubic beziér curve given a point in space. This is another reason why another representation like line segments is much easier to work with in applications like computer graphics.

De Casteljau's algorithm can be used to subdivide a bezier curve recursively by using the properties of bezier curves to calculate new beziér points for both the left and the right part of the old beziér.[Fischer, 2000] A special case of the algorithm is to divide the curve at $t = 0.5$ which gives a first curve $t \in [0, 0.5]$ and a second curve $t \in [0.5, 1]$. An example of this subdivision is presented in figure XXX below, where the first curve is described by the points p_0, p_{01}, p_{012} and p_{0123} and the second curve is described by p_{0123}, p_{123}, p_{23} and p_3 is derived from the initial curve described by p_0, p_1, p_2 and p_3. The following calculations shows how the points is computed.

$$P_{01} = \frac{P_0+P_1}{2}$$
$$P_{23} = \frac{P_2+P_3}{2}$$
$$P_{12} = \frac{P_1+P_2}{2}$$
$$P_{012} = \frac{P_{01}+P_{12}}{2}$$
$$P_{123} = \frac{P_{12}+P_{23}}{2}$$
$$P_{0123} = \frac{P_{012}+P_{123}}{2}$$

3

Related work

3.1 Early EDT algorithms

In a very often mentioned article Danielsson [1980] proposed an improved way to generate distance maps by representing the output of the distance transform as a vector, separating the distance of the different dimensions. Danielsson also proposed two sequential algorithms 4SED (4-point Sequential Euclidean Distance mapping) and 8SED (8-point Sequential Euclidean Distance mapping) for calculating the EDT using his representation of distance. Both the 4SED algorithm and the 8SED algorithm consists of 2 consecutive picture scans where they incrementally update pixel values depending on nearby pixels. The 8SED algorithm is described in the following pseudocode.

```
Data: M: Width, N: Height, L: Binary image of size MxN
Result: L: Distance field
initialization;
for j = 1 to N − 1 step 1 do
    for i = 0 to M − 1 step 1 do
        L(i, j) = min(L(i, j), L(i-1, j-1)+(1, 1), L(i, j-1)+(0+1), L(i+1,
        j-1)+(1,1));
    end
    for i = 1 to M − 1 step 1 do
        L(i, j) = min(L(i, j), L(i-1, j)+(1, 0));
    end
    for i = M − 2 to 0 step 1 do
        L(i, j) = min(L(i, j), L(i+1, j)+(1, 0));
    end
end
```

Algorithm 1: First scan of the 8SED algorithm

```
Data: M: Width, N: Height, L: Binary image of size MxN
Result: L: Distance field
initialization;
for j = N − 2 to 0 step 1 do
    for i = 0 to M − 1 step 1 do
        L(i, j) = min(L(i, j), L(i-1, j+1)+(1, 1), L(i, j+1)+(0, 1), L(i+1, j+1)+(1,
        1));
    end
    for i = 1 to M − 1 step 1 do
        L(i, j) = min(L(i, j), L(i-1, j)+(1, 0));
    end
    for i = M − 2 to 0 step 1 do
        L(i, j) = min(L(i, j), L(i+1, j)+(1, 0));
    end
end
```

Algorithm 2: Second scan of the 8SED algorithm

The two scans are very similar. The only difference is that the first scan is done top down evaluating the pixels below and on the sides of the current pixel and the second scan is done bottom up evaluating the pixels above and on the sides of the current pixel[Ragnemalm, 1993]. The 4SED algorithm and the 8SED algorithm are not error free as Danielsson [1980] proves in his article but he also claims that the errors are rare and small and is negligible for practical purposes.

3.2 Exact EDT algorithms

Distance transforms has been used in different applications in many years. The article from Danielsson [1980] was not the first on the subject but the fact that the algorithms he proposed in his article are some of the most widely used[Fabbri et al., 2008] makes it easy to call Danielsson one of the pionjeers on the subject.

As mentioned earlier in this report, 8SED and 4SED is not exact. Exact algorithms for EDT has only been around since about the 1990s. In an article Fabbri et al. [2008] compares the execution time of exact EDT algorithms. The 6 algorithms compared in the test is Meijster et al. [2000], Maurer Jr et al. [2001], Eggers [1998], Lotufo and Zampirolli [2001], Cuisenaire and Macq [1999] and Saito and Toriwaki [1994]. The conclusion of the comparison is that the algorithms from Meijster and Maurer are the fastest but Meijster's algorithm is preferred due to slightly better performance and the fact that it is easier to implement than Maurer's algorithm.

Meijster's algorithm consist of two separate phases. The first phase iterates through each column performing a distance propagation in both directions. This will create a new image $g(i, j)$ which used in the second phase. The second phase iterates through each row left to right and right to left applying a function $DT(x, y)$ to calculate the output value of each pixel. The function depends on what distance metric is used. The function used for euclidean distance transform follows.

$$DT(x, y) = \min_{0...m}((x - i)^2 + g(i)^2)$$

3.3 Limitations of EDT algorithms

Calculating a distance transform of large size images can be very time consuming. There is a significant difference in computation time between creating a distance map from a 4096x4096 image and creating a distance map from a 64x64 image. For example, the 8SED and the 4SED algorithms both has time complexity O(n) where n is the number of pixels. Time complexity O(n) means that every pixel is visited a constant number of times when transforming the image. Assuming every pixel is visited once and one calculation is done per visit the larger image would require 4096 times more calculations than the smaller image. In an article from Green [2007], distance fields was generated by using a 4096x4096 binary image of a glyph as input to a distance transform. The output from the distance transform was then downsampled to a 64x64 texture. Generating a distance field using a large input image makes the discrete set of possible boundary pixels more dense compared to if a smaller input image is used. The increased density of the possible pixel set helps decrease the calculation error of the distance field assuming subpixel distance measures is not used.

In an article, Gustavson and Strand [2011] showed that the calculation errors of distance transforms can be decreased by using information about the subpixel boundary between the foreground and background pixels. In the article they use an anti-aliased input image to be able to locate the boundary in the pixels to get more precise distance measures between the pixel and the boundary. They show that they get approximately the same amount of errors using their method on a 16x16 times smaller input image than by using the method Chris Green

proposed in his article. The smaller input image increases the the execution time and decreases the memory consumption by a factor of approximately 30 times according to Gustavson and Strand.

4

Method

This chapter will give a detailed description on which how the problem real-time textrendering on mobile devices was solved in this thesis. This chapter is divided into two different parts. The first part gives a detailed description about the implementation of the distance transform module. This part has been reimplemented several times due to poor performance. The second part gives a detailed description about the implementation of the distance field rendering module.

5 Resultat

Det här är kapitlet där resultaten presenteras.

5.1 Ditten

Liksom [**?**] har vi kommit fram till att glass smakar bäst på sommaren.

När vi nu går in på hur glass smakar vid olika tidpunkter under dagen hänvisar vi till figur 5.1, och speciellt till figur 5.1b. Jämför sedan med figur 5.2 för att se hur det kan bli när man äter glass vid okontrollerade tidpunkter.

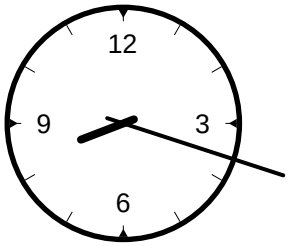
Veselić, Krešimir (Veselić, Krešimir) skrev en gång en artikel med titeln *Bounds for exponentially stable semigroups*.

Kommer att tänka på en liten anekdot...

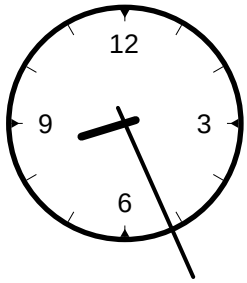
TODO:
Ta bort den löjliga anekdoten!

5.2 Framtiden

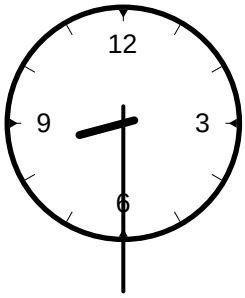
Sen när glassen är uppäten är det bara till att sätta igång och skriva på exjobbet igen!



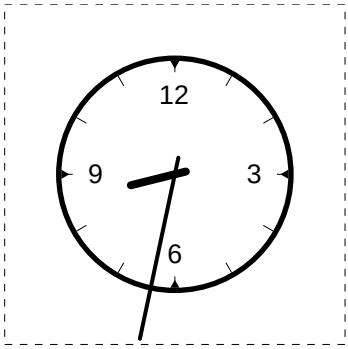
(a) *Det här är väl tidigt — din glass hinner smälta innan ditt sällskap dyker upp.*



(b) *Kiosken stänger snart, men inte nu — perfekt!*



(c) *Precis i tid — du får in ett finger i luckan just när kiosken ska stänga. Han som jobbar blir sur, och det blir smolk i bägaren.*



(d) *Du är sen — kiosken är stängd.*

Figure 5.1: *Illustration av* subfloats. *Den så kallade* bounding box*en visas i (d). Lägg märke till att bounding boxen har satts så att alla bilder har samma storlek, med enhetlig placering av själva innehållet i förhållande till bounding boxen. Antag att du ska träffa en kompis för att äta glass just när kiosken stänger för dagen vid 08:30. När dyker du upp?*

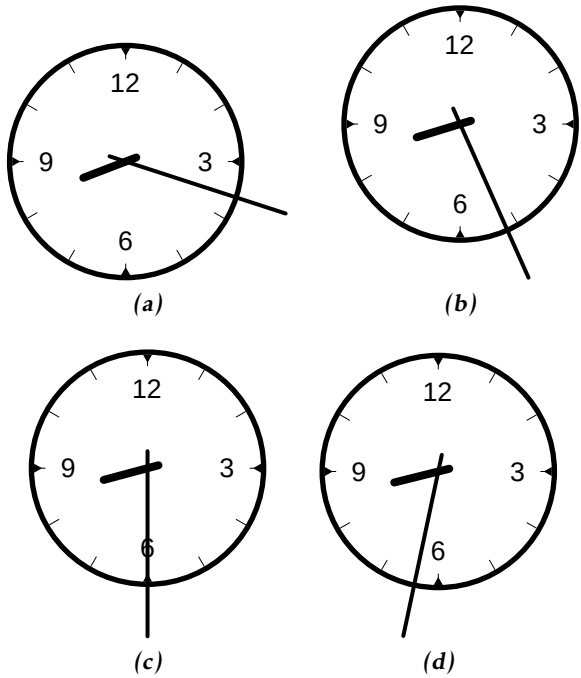


Figure 5.2: *En andra illustration av* subfloats. *Den här gången har bounding boxen gjorts så liten som möjligt runt själva innehållet. Resultatet är stökiga placeringar på sidan. Samma sak kan hända med vanliga fyrkantiga figurer när man har text som spretar ut åt lite olika håll från själva rutan med kurvor i.*

Appendix

5.A Ett par långa bevis

Det här är en appendix-del av det aktuella kapitlet.

6

Avslutande kommentarer

Sätt av ett kort kapitel sist i rapporten till att avrunda och föreslå rikningar för framtida utveckling av arbetet.

Appendix

A

Trista saker

Långa beräkningar brukar bli rätt trista…

Detta är ett appendix-kapitel. Jämför med appendixet i kapitel 5.

A.1 Bädda sängen

Den här beräkingen är så trista att vi kallar den *att bädda sängen*.

A.2 Diska

Den här beräkingen är så trista att vi kallar den *att diska*.

B

rtthesis documentation and LaTeX tips

This document is not only an example that you can use to get started with the *rtthesis* class, it also contains written instructions for how to use the class, and some general tips on how to use LaTeX to produce a beautiful thesis. As we do so in this chapter, we also get the opportunity to look at some theorem-like environments, which you can alter the look of by changing the options given to the *rtthesis* class.

B.1 Basic setup

You must decide on an input encoding from start, and select the corresponding class option from tabell B.12 on sida 40. You must also tell *rtthesis* whether you intend to use part sectioning or not, see tabell B.1. There are many more class options, but they will be mentioned below where there is room for a more detailed discussion for the corresponding features.

Information about the thesis, which is needed to produce the thesis itself as well as the thesis cover and the "spikblad", is passed to *rtthesis* using the command `\setupThesis`. The command is called in the following way, where the most common key-value pairs are listed in tabell B.2 (the remaining key-value pairs concern master's theses, see avsnitt B.22)):

```
\setupThesis{
  key1=value1,
  key2=value2,
  ...
}
```

If a PhD thesis has an interesting illustration on the cover, it is customary to provide a caption for the illustration. The caption will be printed on the back of the title page, and is set up by redefining the command `\rtcoverinfo`. For instance, it may look like this:

Table B.1: *Class options that inform* rtthesis *whether part sectioning will be used or not.*

Class option	Meaning
`parts`	Prepare for `\part` as the topmost sectioning command.
`noparts`	Prepare for `\chapter` as the topmost sectioning command.

Table B.2: *Key-value pairs recognized by* `\setupThesis`*. Note that values that include white space are surrounded by braces.*

Key = **Example value**	Comment
`author = {My Name}`	
`title = {Thesis title}`	
`subtitle = {Good stuff}`	Optional.
`city = Norrköping`	Default: *Linköping*
`year = 2010`	
`isbn = isbn-isbn-isbn-isbn`	
`type = phd`	Must be either *phd*, *lic*, or *msc*.
`thesisNo = 9999`	Number in series (the series is determined by the choice of thesis type).
`localID = 11`	Only used for licentiate's theses. It is the last part of the local identifier *LIU-TEK-LIC-2010:11* in this case.
`username = isyusername`	Used to generate the author's email address.
`dedication = {To my parents!}`	

```
\renewcommand{\rtcoverinfo}{\textbf{Cover illustration:} Block
diagram showing the structure of the control scheme proposed in
\chapterref{cha:cool-control}}
```

B.2 Page layout and related options

Theses are restricted to the S5 paper size. How the S5 page is organized is up to you, but *rtthesis* only allows you to choose from two predefined layouts, and only one of them is recommended. To get your own layout you should make a copy of *rtthesis.cls* and modify the code for one of the existing class options for layout. The class options for page layout are given in tabell B.3.

At the time of writing, the printers used by LiU-Tryck print on A4 paper (physical size), which is then cropped to S5 (logical size). Similarly, when you print draft versions of your thesis on your office printer, it is very likely that the used physical paper size will be A4. Hence, it makes sense to let *rtthesis* control how

Table B.3: *Class options related to page layout. The most important one to remember is* `crop` *(since* `S5` *and* `pdf` *are default).*

Class option	Meaning
`S5`	Recommended layout. Margin paragraphs are tiny (see avsnitt 5.1 for examples), and should only be used for comments that will be removed in the final version of the thesis. Default.
`S5MP`	Layout to use if you are serious about margin paragraphs. Not recommended, since the S5 format is too narrow to really fit margin paragraphs of reasonable width.
`nailing`	Layout for the "spikblad". Not for theses!
`pdf`	Produce pages in the S5 format. Default.
`onA4`	Logical S5 page on a PDF page of size A4.
`info`	Write information about each page above the logical S5 page.
`crop`	Same as `onA4` with `info` and crop marks.
`noInfo`	Turn off the effect of `info`.
`draft`	Same as `onA4`, but pictures are blank and overfull `hbox`es stand out.

the S5 logical page is placed on the A4 physical paper. In this case, *rtthesis* will produce a PDF with pages in the A4 format, with content restricted to the S5 format. On the other hand, when you produce a PDF that is meant to be read on a computer screen, the page size should be exactly S5. When targeting the A4 physical format, it is possible to get crop marks for the S5 box, and to put some information about each page outside the S5 box. The related class options are given in tabell B.3.

To ensure that you really get the page layout you think when you send your thesis file to the printer's, the best option *should* be to use the `crop` option. However, they will tell you differently, since they think it's *their* job to position the logical page on A4 and add crop marks. Unfortunately, there is a lot of manual work in the process, so there is a (substantial?!) risk that the content of your pages will be shifted with respect to the S5 box of your layout...

Although only weakly related to page layout, this section ends with a tip for how to change the size of the chapter numbers (some users find them much too big). The font is controlled using the sectsty package, and it follows that it can be redefined by, for instance,

```
\chapternumberfont{\fontsize{60mm}{63mm}\selectfont}
```

Table B.4: *Legal option values to the* `notation` *environment. The options control the look of the* `notationtabular` *environments used inside the* `notation` *environment. The initial definition of* `notationtabular` *is the same as that obtained by passing the option* `new`*.*

Option	Meaning
emty	Do not redefine `notationtabular`. Default.
`old`	Make `notationtabular` produce a plain LaTeX table with double horizontal lines under the table headings, and a vertical line separating the two columns.
`new`	Make `notationtabular` produce a table according to the guidelines in Mori [2007] using the ctable package.

B.3 Front-matter environments

There are environments defined for typical sections in the front-matter[1]. The most important purpose of providing these environments is that they take care of the table of contents and the PDF bookmarks for you. The environments are `abstract`, `preface`, `acknowledgments`, and `notation`.

The environment `abstract` accepts the language used inside the environment as an optional argument (which defaults to `english`). If the language is set to `swedish`, the title of the abstract will be *Populärvetenskaplig sammanfattning*, in accordance with the Linköping University requirements on theses written in English.

Inside the `notation` environment, you can put anything you like, and maybe the `notationtabular` environment provided by *rtthesis* suits your needs. In order to define this environment, *rtthesis* loads the two packages array and ctable, and also defines the command `\otoprule` to mean the same as `\toprule`. See tabell B.4 regarding how to change the look of `notationtabular`.

There is a class option called `noextras`, which was intended to inhibit the effect of the `\maketitle` command, and redefine the front-matter environments to not produce any output. However, the option is not working well at the moment. On the other hand, as the time it takes to compile a thesis on a modern computer is very short, it is rather unclear why someone would like to use this feature anyway.

[1] The *front-matter* is everything that goes in the beginning of the thesis, before the page numbered *1*.

B.4 Abbreviations

Automatic control is a LaTeX-friendly community. This means that everything you produce is expected to look good. We begin with a basic result.

Theorem B.1. *Abbreviations, such as ARMA, look best in small caps.*

Proof: Just compare with "ARMA". □

However, it is important that the small caps match the sorrounding text, compare the statement in the theorem above with the following variation of it, in italics instead of slanted text:

> *Abbreviations, such as* arma[2] *or* ARMA, *will stick out in a terrible way if you don't watch out!*

This is why the *rtthesis* class uses slanted text rather than italics in theorems rather when slanted small caps are available.

Unfortunately, *rtthesis* does currently not provide a way to make small caps look good in italics, which leads to the following corollary to teorem B.1.

Corollary B.2. *One has to make a choice between*

- *Beautiful abbreviations using small caps (instead of ordinary upper case).*
- *Pretty text typeset in italics (instead of slanted text).*

B.5 Definitions

Let us discuss another theorem-like environment while we have some examples of similar environments to compare with in the previous section. That is, let us discuss the `definition` environment (and the similar environments `assumption` and `remark`). All the theorem-like environments are defined in a separate package, rtthesis-theorems, so that they can be used with other document classes as well. The definition below is an example of a definition with a title.

Definition B.3 (Definition). A *definition* is a precise explanation of the meaning of a word or concept. It may be tempting to include examples in a definition, but a good definition should not depend on examples as part of the definition. However, examples are often useful to clarify a definition, and should appear near the definition.

A short definition may require just a single paragraph, while a more complex definition may require a few paragraphs. Some definitions will also make use of displayed math.

One problem one has to consider if definitions are not restricted to just one paragraph, is how to show the reader where the definition ends. In theorems, it is common to use italics or slanted text (for brevity, we will not mention italics

[2] This will cause a LaTeX warning.

from here on) to show where the theorem statement ends, but for definitions it may be desirable to use the slanted text to emphasize the word or concept being defined. (It is arguably more clear to highlight the new word or concept using slanted text with upright surrounding text, than vice versa.) To use an upright font for the definitions may also be a way of avoiding to heavy use of slanted text.

Various options related to the appearance of theorem-like things (in LaTeX, a definition is a kind of theorem) are described in tabell B.5. Tabell B.8 (used also to illustrate tables) contains some suggestions regarding combinations of options for the `definition` environment and options for paragraph breaks.

Sometimes, a definition may be given without a title. The next definition is an example of this, even though it is questionable whether it was a good idea to omit the title in this particular case.

Definition B.4. An *environment* in LaTeX is a construct that is entered with the command `\begin{...}` and exited with the command `\end{...}`, where "..." should be the name of the environment.

In tabell B.5, there are three options related particularly to how `definition`, `assumption`, and `remark` are typeset.

- With `definition=naked` (default) the definitions are typeset in upright font, and there is nothing on the page that marks the end of the definition.
- With `definition=theorem` the definitions are typeset in the same style as theorems. Since theorems are supposed to be typeset in slanted text, this will make it clear where the definition ends.
- With `definition=marks` the beginning and end of definitions will be indicated with small marks. Compare how the end of a proof is marked with a square box! The current implementation has some problems with placing the marks if the definition ends with a displayed equation, but this can be compensated for by manual insertion of a `\vspace` command.

You may judge from the following example whether manual insertion of a `\vspace` command is necessary to make the definition ending with a displayed equation look alright.

Definition B.5. The factorial (denoted by the postfix operator !), defined for natural numbers, is given by

$$n! = \begin{cases} 1, & \text{if } n = 0 \\ n \cdot (n-1) \cdot \ldots \cdot 1, & \text{otherwise} \end{cases}$$

This paragraph only serves to highlight the vertical white space below the definition ending with a displayed equation. Note that one way to avoid problems with this kind of definitions is to rewrite them so that they don't end with displayed equations.

All definitions in this section have been entered as isolated paragraphs; that is, there is an empty line in the source code of the document before and after each

Table B.5: *Class options related appearance of theorem-like environments. The* theorem-like environments *defined by* rtthesis *are* `theorem`*,* `proposition`*,* `lemma`*,* `corollary`*,* `definition`*,* `assumption`*, and* `remark`*. The* definition-like environments *are a subset of the* theorem-like environments*, consisting of the environments* `definition`*,* `assumption`*, and* `remark`*. See also tabell B.10 regarding the fonts used in theorems.*

Class option	**Meaning**
`break`	Put line breaks after the titles of the environments `theorem`, `proposition`, `lemma`, and `corollary`.
`nobreak`	Never put line breaks after titles of theorem-like environments. Default.
`definition=naked`	Definition-like environments look like the surrounding text, and are only isolated by some vertical white space. Default.
`definition=theorem`	Definition-like environments use same font as the `theorem` environment, and are isolated by some vertical white space.
`definition=marks`	Definition-like environments look like the surrounding text, and are isolated by small marks. Strongly recommended if `parskip` is used.
`nosharecounter`	Use separate numbering sequences for each theorem-like environment and the `example` environment.
`sharecounter`	Use one numbering sequence for theorem-like environments, and the `example` environment.

`definition` environment. Although not recommended, *rtthesis* supports definitions that are connected with the preceding paragraph, in which case the usual vertical space (if any) between paragraphs will not be inserted. *Be careful so that you don't omit the paragraph breaks by mistakes, since it makes a difference that may be hard for proofreaders to spot!* As an example of a definition written in the same paragraph as the preceding text,

Definition B.6. A *paragraph* (according to Oxford American Dictionaries) is a distinct section of a piece of writing, usually dealing with a single theme and indicated by a new line, indentation, or numbering.

There is no paragraph break in the source code between the definition above and this text, but currently this cannot be seen in the typeset document. If you know how to solve this, let the *rtthesis* maintainer know! If you want to learn about the TeX mechanisms involved, see Ryćko and Jackowski [1993].

B.6 Theorem titles

The class lets you control the white space that separates a theorem title from the theorem statement. The options appear in tabell B.5. With the class option `break` (default), you will get a line break. With `nobreak`, you will just get horizontal space. Not all types of theorem-like environments will be affected by the `break` option, so to get things exactly they way you want, you may have to make your own modified copy of the *rtthesis* class. Try to recompile the document with the two different options and compare the result!

B.7 To share or not to share counters

Other things to think about regarding style include whether to use the same counter for all sorts of theorem-like things. Again, the options appear in tabell B.5. Some like to make the number of important theorems to stand out by having a separate counter (as in Khalil [2002]), while other prefer to use as few counters as possible in order to make it easy to locate referenced items (as in Rugh [1996]). The two alternatives are supported in *rtthesis*, via the options `sharecounter` and `nosharecounter`.

B.8 Completely customized theorem-like environments

If you don't like the way *rtthesis* sets up theorem-like environments (listed in the caption of tabell B.5) for you, you may pass the class option `notheorems`. Then amsthm will not be loaded, none of the theorem-like environments will be defined, and it is up to you to define your own environments. If you decide to do so, using the amsthm package will be a good idea.

Table B.6: *The lengths used to control the appearance of the* `example` *environment. Note that the environment tries to compensate for the current value of* `\parskip`*, so you may not always get exactly what you'd expect. Also, the meaning of the distance between the upper stroke and the text is somewhat arbitrary in order to allocate space for the example title.*

Length	Default	Purpose
`\exampleLineWidth`	0.6 pt	Thickness of the strokes.
`\exampleTopBotInnerMargin`	2 ex	Vertical space between strokes and contents of the example.
`\exampleTopBotOuterMargin`	1 em `plus` 1 ex `minus` 1 ex	Vertical space surrounding the example.

B.9 The `example` environment

The `example` environment defined by the *rtthesis* class is *not* a floating environment, but is simply used to highlight that the text inside the environment is just an example of something more general that you have explained before. Just as with the theorem-like environments, the environment is defined in a separate package, rtthesis-example, so that it can be used with other document classes as well.

Example B.7

As an example of the `example` environment, we include a little example here. You can use this example to see how the options described in avsnitt B.7 affects the numbering of the environment.

Depending on where this example ends up in the typeset document, you may also have the chance to see the ugly stretched vertical space that sometimes appears at the top and bottom of the environment.

There are three lengths you may play with the fine tune the appearance of examples, explained in tabell B.6. Clearly, it would be possible to introduce additional parameters, but currently the corresponding aspects of the environment are hard-coded into *rtthesis*.

As is mentioned in the example above, there is sometimes problem with vertical space at the top and bottom of the `example` environment. During the page breaking process (see avsnitt B.17) you could consider to add something like

```
\vspace{-1\baselineskip}
```

to reduce such artifacts. Even better, if you know how to correct this in the definition of the environment, let the *rtthesis* maintainer know! The paper Ryćko

and Jackowski [1993] is recommended for anyone interested in the lesser known details of TeX that one has to grasp in order to really solve the problem.

B.10 Captions

The *rtthesis* class loads the captions package to obtain good-looking captions. Captions are set up assuming that table captions will be placed above the table they belong to. Many authors find this confusing since figure captions are always placed below the figure they belong to. If you want to put table captions below the table you need to adjust the spacing around the caption by putting the following line in your personal style file:

```
\captionsetup[table]{position=bottom}
```

Note that the command above only changes the spacing around the caption. You still have to put the code for each caption relative to the tabular itself consistently with the captions setup. Two tables are included in this document for illustration. Tabell B.8 indicates the many combinations of options that the `definition` environment has been designed to work with. The next one, tabell B.7 is just a stupid table telling where the different chapters in this document begin. For comparison, a typical automatic control block diagram has been included in figur B.1.

Some nice guidelines for table creation in LaTeX are given in Mori [2007] (it is just two clicks away!).

B.11 Hyperlinks

For readers our the electronically published version of your thesis, as well as yourself while your are working on it, it is very convenient to have working hyperlinks in the document.

B.11.1 Basic setup

Basically, hyperlinks are obtained by using the hypreref package. However, this package has quite a lot of compatibility issues with other packages, and knowledge about how to deal with these issues is coded into the *rtthesis* class. That is, all you should have to do to get hyperlinks in your document is to specify the `hyperref` option to *rtthesis*. The class options related to the linking infrastructure of the document are listed in tabell B.9.

At the time of writing *rtthesis* does not call `\hypersetup` with information about document title, keywords, and other information provided to `\setupThesis` (see tabell B.2). If someone wants this, it shouldn't be hard to do.

B.11.2 Hyperlinks and electronic publishing

To make your dear hyperlinks survive all the way to the electronic publishing system, you may have to replace the file that is sent to e-press by LiU-tryck. The

Table B.7: *Different combinations of class options that affects the* `definition` *environment. The code for this caption appears at the beginning of the* `table` *environment. It would have had the desired distance to the tabular if the default caption setup of* rtthesis *was used, but this document has been set up for table captions below the corresponding tabular.*

Chapter	Title	Page
1	Introduction	1
5	Resultat	13
B	*rtthesis* documentation and LaTeX tips	23
A	Trista saker	21

Table B.8: *Different combinations of class options that affects the* `definition` *environment. The code for this caption appears at the end of the* `table` *environment. It will be too close to the tabular using the default settings of* rtthesis *(but note that this document has been setup differently, see avsnitt B.10).*

	definition=		
	naked	**theorem**	**marks**
noparskip	OK	Avoid	OK
parskip	Bad	Avoid	OK

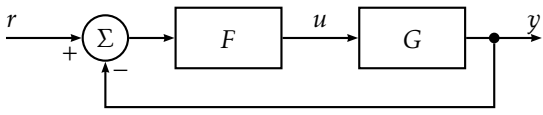


Figure B.1: *A simple illustration in a floating* `figure` *environment. Note that figure captions are always placed under the corresponding figure, and hence that the caption code should always appear at the end of the* `figure` *environment.*

Table B.9: *Class options related to (hyper) linking infrastructure.*

Class option	Meaning
`hyperref`	Turn on hyperlinks using the hyperref package. Default.
`nohyperref`	Turn off hyperlinks, and compensate for commands no longer provided by the hyperref package.
`backref`	Turn on bibliography back references. Default.
`nobackref`	Turn off bibliography back references. (Currently required if you plan to use the features of bibunits.)

problem is that LiU-tryck creates a compressed version of the file that is used in the printer, and the compression will remove nice features such as page numbers, hyperlinks, and bookmarks. Fortunately, the guys at e-press seem to be understanding and will accept to publish a file that they receive directly from you.

B.11.3 Page number formatting in the index

If you use an index in your thesis, you will often want to change the formatting of certain page numbers in the index. Without hyperref, this could look like

```
hyperlinks\index{hyperlinks|textit}
```

to get the page number for this occurrence of *hyperlinks* to be typeset in italics. The problem with this is that this page number will not be a hyperlink, while other page numbers will be hyperlinks to the correct page. To get both italics and a hyperlink you need to define a special index formatting commands like the following.

```
\newcommand{\hyperpageit}[1]{\textit{\hyperpage{#1}}}
\newcommand{\hyperpagebf}[1]{\textbf{\hyperpage{#1}}}
\newcommand{\hyperpagefootnote}[1]{\hyperpage{#1}n}
```

Now, you can write

```
hyperlinks\index{hyperlinks|hyperpageit}
```

to get both italics and a hyperlink. The *rtthesis* class will provide a trivial definition of `\chapter` in case hyperref is not loaded, so you may safely start to use the above definitions even if you are not sure whether you will use hyperlinks in the end.

B.11.4 Friendlier hyperlinks

The default mechanism for references in LaTeX, being the command `\ref`, is modified as expected by the hyperref package. For instance, the number in “chapter B”

is linked to the beginning of the current chapter (if you click it, be sure to just the *jump back* function of your PDF viewer to get back to here!). However, all of "this" is also a link to the same place. That is, it is possible to other things than the number itself as links. We could also make a reference that will never be linked, like in "chapter B".

So, what's so friendly about this? What I'm aiming at is that you can say "chapter B". The code for this link is

```
\hyperref[cha:rtthesis]{chapter~\ref*{cha:rtthesis}}
```

Of course, it is very annoying to repeat the key twice; first to point the hyperlink to the correct place, second to show the number of the chapter. With the `\autoref` command from the hyperref bundle, we get "Appendix B". This is almost perfect. The problem is that one cannot get an uppercase initial at the beginning of a sentence without redefining "chapter" to "Chapter",

```
\renewcommand{\Chaptername}{Chapter}
```

but then we will not get the nice lower case initial in the middle of a sentence. Many authors don't bother about this and use uppercase initials irrespectively of where in a sentence the reference appears.

The only solution I (Henrik Tidefelt) knows of, is to define special commands for each type of reference. A basic solution might look as follows.

```
\newcommand{\chapterref}[1]{\hyperref[#1]{chapter~\ref*{#1}}}
\newcommand{\Chapterref}[1]{\hyperref[#1]{Chapter~\ref*{#1}}}
```

You should then use `\chapterref` in the middle of a sentence, and `\Chapterref` at the beginning of a sentence. I you later decide that you want to have upper case initials everywhere, you just have to change your definitions to

```
\newcommand{\chapterref}[1]{\hyperref[#1]{Chapter~\ref*{#1}}}
\newcommand{\Chapterref}[1]{\hyperref[#1]{Chapter~\ref*{#1}}}
```

A more complete solution will also provide commands for the plural forms "chapters" and "Chapters".

It is also nice to use a similar technique for page references. For instance, this chapter starts on page 23, and such links can be created easily using a command like

```
\newcommand{\pagepageref}[1]{\hyperref[#1]{page~\pageref*{#1}}}
```

Because of the many possible preferences for how to handle labels and references within documents, *rtthesis* does not define any related commands. The current section should give you some ideas of what can be achieved, and now it is up to you to design your own solution or borrow a solution from someone else (or simply stick with `\autoref` or the 1980's way of doing things)!

B.12 Backreferences from the bibliography

By default, *rtthesis* uses the backref package to put references from the bibliography back into the text. The options for turning this feature on and off are listed in tabell B.9.

By controlling this feature via the class, the choice whether to use it or not can be made orthogonal to the choice of whether to use hyperref or not.

In addition to just loading backref, *rtthesis* will do a basic setup of the commands used to typeset the list of page numbers for each reference. This behavior can easily be redefined without modifying the *rtthesis* class file. See the backref documentation for details on how to do this!

B.13 Using the bibentry package

The bibentry package makes it possible to use the information in the bibliography to present your publications at any place in the document. In order to work independently of whether you use back references from the bibliography or not, you need to follow the pattern below each time you use the `\bibentry` command, where `KEY` is the same key to you publication that you would with use with any other citation command.

```
\begin{quotation}
  \nocite{KEY}\noindent
  \backrefparscanfalse\bibentry{KEY}.\backrefparscantrue
\end{quotation}
```

To use the `quotation` environment is just a suggestion — it will make the reference stand out by using a some what shorter text line width. Note the period that follows the `\bibentry` command — the command leaves it up to you how to terminate the entry. The `\nocite` command ensures that the reference appears in the bibliography, which is necessary to produce the entry. The `\noindent` commands simply prevents the first line in the `quotation` from being indented. The commands `\backrefparscanfalse` and `\backrefparscantrue` are related to the backref package used to produce back references from the bibliography, and should always surround the `\bibentry` command. In case you have turned back references off using the `nobackref`, *rtthesis* will provide substitutes for these two commands.

B.14 Fonts

Though basically not a task for a LaTeX class, *rtthesis* will assist in loading some font packages. There are some class options that control this behavior, described below, and if these options are not good enough for you, you may have to make your own copy of the class and replace the font packages you don't like. Options for font selection are listed in tabell B.10.

One reason, however, for letting *rtthesis* handle the font selection is that this makes it possible for the class to do some things more intelligently. At the mo-

Table B.10: *Class options related to fonts. When slanted small caps are activated, theorem-like environments will use slanted text instead of italics. The lower part of the table are examples of options that will be understood by the kpfonts package, and are only meaningful in combination with the* `kp` *option. (Note that options passed to* rtthesis*, but that are not understood by* rtthesis *will be passed on automatically by LaTeX to loaded packages.)*

Class option	Meaning
`kp`	Use KpFonts (Kepler) and activate slanted small caps. Default.
`times`	Use Times and deactivate slanted small caps.
`lm`	Use Latin Modern and deactivate slanted small caps.
`largesmallcaps`	Let the small caps be slightly higher than an *x*. See the KpFonts documentation!
`intlimits`	Placement of integration limits. See the KpFonts documentation!
`widermath`	Put just a little more horizontal space between entities in math mode. See the KpFonts documentation!

ment, *rtthesis* will help you make use of some of the goodies of KpFonts, if you choose to use that font.

B.15 Hanging punctuation

The *rtthesis* class automatically loads the pdfcprot package with its default settings. It uses a pdfTeX feature to make punctuation hang into the right margin. If you don't like it, make your own copy of the class and comment out the line that loads the package. One reason not to use it would be if your document will be (perhaps only occasionally) typeset using the old TeX program, since this will lead to noticeable differences in the line breaks compared to when pdfTeX is used. No matter what you choose, make your choice *before* you start working with the page breaks in your document!

B.16 Paragraph breaks

There are two common ways of visualizing paragraph breaks in a document, illustrated by the two examples below. The look of paragraph breaks is controlled using the class options listed in tabell B.11.

Table B.11: *Class options related to formatting of paragraph breaks.*

Class option	Meaning
`noparskip`	US style, see exempel B.8. Default.
`parskip`	European style, see exempel B.9.

Example B.8: Indented first line

This style is still the most common. It is particularly dominant in text written in the US.

It is a matter of style whether to omit the indentation of the first line after a sectioning command such as `\chapter` or `\subsection`. The omission is typically automated, but can also be enforced using the `\noindent` command.

One drawback of not having vertical space between paragraphs is that it will be harder for pdfTeX to find good places for page breaks, compared to the option shown below. If you like compact documents, however, this is the option for you!

For testing purposes, this example ends with a paragraph break, so that TeX is in *vmode* at the end. You should always avoid this, but the class will try to compensate for your mistakes...

Example B.9: Vertical white space

This style is still increasing in popularity. It is rather common in modern texts written in Europe, and the style has received special attention from the Netherlands TeX user group *Nederlandstalige TeX Gebruikersgroep, NTG*. Their efforts can be used through their variants of the standard LaTeX classes.

Unfortunately, the NTG classes are not compatible with *rtthesis*, and the solution provided by the parskip package is only part of the solution. Hence, *rtthesis* will do more than just loading the parskip package for you if you specify the `parskip` option.

A good reason to put code related paragraph breaks in the class file is that all the small adjustments that different people come up with can be put in one placed so that they are accessible to future users of the class.

B.17 Page breaks

There is a whole lot to say about how to obtain nice page breaks. You will find some recommendations below, but do not use this document as your ultimate reference on this topic! (This document itself contains some really nasty page breaks — at least at the time of writing this — as a result of not paying any attention at all to the problem. It would simply bee too time-consuming to keep adjusting the page breaks each time the document is edited.)

- Take no consideration of page breaks until page breaking is the only aspect of your thesis that remains to be taken care of! Page breaking involves a lot of manual intervention of the automatic mechanisms in pdfTEX, and as soon as you have started to intervene, any further changes to the text will risk to ruin your page breaking fixes, and may even lead to worse results than before since the automatic page breaking has been tampered with.

- First thing to try is to make changes to the text to help the automatic page breaking mechanism. Try to make sentences longer or shorter depending on the situation. Since this will not tamper with the automatic page breaking mechanism, this option will incur the least loss of maintainability of your document.

- Can the location of floats be changed to improve page breaks? Play around with exactly where in your source files the code for the floating environments appears!

- You may also try to force early page breaks using the `\Needspace*` command. For instance, putting

  ```
  \Needspace*{2\baselineskip}
  ```

  before a paragraph will cause a page break if there is not enough vertical space on the page to hold two lines of text. The good thing about this option is that your intervention will cause no harm if the `\Needspace*` command appears in the middle of a page. The bad thing about this option is that it may cause remaining vertical space on the broken page to be stretched quite badly. You should always check that the resulting page looks OK!

  For more information, and related commands, see the documentation for the needspace package!

- The last option is to play with the vertical size of individual pages. For instance, putting

  ```
  \enlargethispage{2\baselineskip}
  ```

  before a paragraph you would like to fit into the current page will make space for two extra lines of text. This avoids the bad stretching of vertical space that the `\Needspace*` option may cause. However, if you would make other changes that makes tampering with the page size unnecessary, it will be very time-consuming to detect this and remove the no longer needed `\enlargethispage` command.

Note that manual page breaking is a time-consuming task. Make sure to have at least one full day allocated to page breaking before you submit your thesis for print!

Table B.12: *Class options related to input encodings. Note that there is no default;* rtthesis *requires one of these options to be passed explicitly.*

Class option	Meaning
`latin1`	Simply use inputenc with option `latin1`.
`utf8`	Use inputenc with option `utf8`, and define some additional characters.

B.18 Input encoding

Two input encodings are supported, being latin-1 and UTF-8. The choice of input encoding should be made via the *rtthesis* class, so that the class can use the correct encoding to define certain global strings. The input encoding options are listed in tabell B.12.

Choose latin-1 if you depend on lots of files using this encoding, and do not want to change the encoding of these files. Changing the encoding of a file is easy both in Emacs and using the *iconv* command line utility. The latin-1 encoding is the default in *rtthesis*, but the choice can be made explicit by passing the `latin1` option to the class.

Choose UTF-8 to be able to type many more characters directly in your LaTeX sources compared to latin-1. For instance, names of foreign authors often use characters that cannot be entered directly using latin-1. In UTF-8, most of these as well as special punctuation characters such as double quotes and various dashes can be entered directly in the source. Use the `utf8` class option if your files are encoded in UTF-8.

The current implementation of UTF-8 in the inputenc package only defines the input encoding for characters that have corresponding glyphs in active fonts (see the inputenc documentation for details). This means that some characters that TeX would build by combining several glyphs will not be defined by inputenc. If the `utf8` is given, *rtthesis* will define a list of additional characters by inclusion of the package rtthesis-utf8-ext. If you need additional characters, you should make your own package similar to rtthesis-utf8-ext, and then let the maintainer of *rtthesis* know, so that the additional characters may be added to rtthesis-utf8-ext so that others can use them in the future. Note that rtthesis-utf8-ext may be a useful package also when you are not using the *rtthesis* class.

It is easy to set up Emacs so that it uses the UTF-8 encoding for your TeX files, but it is out of the scope of the current document to give further explanations here.

B.19 *rtthesis* and natbib

Interoperability with different bibliography packages is a tricky issue. It has been a design decision to try to support at least natbib, at the cost of loosing compatibility with other packages such as jurabib. The core of the problem is package loading

Table B.13: *Class options related to the natbib package. Note that options can be passed to natbib by passing them as global class options to* rtthesis. *See the natbib documentation for more useful options.*

Class option	Meaning
`authoryear`	Default option of natbib — no need to specify.
`round`	Default option of natbib — no need to specify.
`colon`	Default option of natbib — no need to specify.
`square`	Example of option that natbib will pick up (alternative to `round`).
`comma`	Example of option that natbib will pick up (alternative to `colon`).
`numbers`	Conflicting natbib option — forbidden in combination with `usebibunits`, see `forcenumbers` below.
`forcenumbers`	Enforce option `numbers` to be passed to natbib (alternative to `authoryear`) — it's up to you to resolve the conflict.

order, requiring natbib to be loaded very early on in the class. To pass options to natbib, pass them as global class options to *rtthesis*. Note that the default options for natbib are quite reasonable, and see tabell B.13 for examples of other options that natbib will pick up. If you know how to resolve the conflict with the natbib option `usebibunits`, let the *rtthesis* maintainer know!

B.20 The lists of previous theses

The lists of previous licentiate's and PhD theses can be found in *liclist.tex* and *phdlist.tex*, respectively, and the appropriate one of the is automatically included at the end of your thesis. Both files are found in the directory *$TEXMFGROUPLOCAL/tex/latex/rt/rtthesis* .

Note that it is *your responsibility* to make sure that your thesis is added to the appropriate list after you have sent it to print but before the next thesis of the same kind is printed. If other people are writing theses at the same time as you, you will have to coordinate your moves in order to make sure that the lists get updated in the correct order. To get your thesis added to the appropriate list, you simply send an email with information about your thesis to the *rtthesis* maintainer. The information shall be in one of the following formats:

```
\licitem{J.~Doe}{Title}{Thesis No}{YYYY}
```

or

```
\phditem{J.~Doe}{Title}{Theis No}{YYYY}{ISBN}
```

It is a good idea to make a copy of the file you need when it is time to print. If you don't make a copy, and then compile your thesis again at a later time, the list will be wrong because it will include at least one thesis that wasn't prior to yours — namely your own!

B.21 Compilation theses

The *rtthesis* class aims to support the production of both monographs and compilation theses. There is a compilation thesis example included with *rtthesis*. Please have a look at that while reading the sections below!

B.21.1 Including publications in your thesis

It is assumed that included publications shall be compiled together with the rest of your thesis, as opposed to being included as exactly the way the look where published. Under this assumption, it is reasonable to expect things such as a suitable chapter numbering, and that the global table of contents includes the sections withing publications. Note that it would be rather difficult to get things such as the table of contents and other infrastructure right if publications were to be included by direct PDF inclusion.

The `papers` environment provided by *rtthesis* will redefine commands and set up some additional commands to support the inclusion of LaTeX sources of your publication. It is recommended that the environment is placed in a second part of the thesis. Inside the environment, the `\chapter` command is redefined to both start a new chapter and set up the title of the publication to be included in the same chapter. Chapters will be labeled with letters instead of numbers, so it is up to you to make a clear distinction between referencing an appendix chapter and a publication chapter.

If the title of a publication is too long to fit in the page header, you may follow the `\chaptermark` command by a `\chaptermark` command. Since the `\chaptermark` command takes an optional argument to be used in the table of contents, there are three different variations of the publication title that can be defined.

The word for publications used by *rtthesis* is *paper*; it will appear both on the chapter title page and in page headers. To change this to something else, you simply have to redefine `\chaptername` to something else inside the `papers` environment.

After setting up the publication title, the `\author` command should be used to set up the list of authors. It works as usual, but sports two special *rtthesis* commands that should be used when there are two author affiliations; put `\authorleft` immediately after author names who's affiliation should appear to the left below

the list of authors, and put `\authorright` after the other authors. There is currently no support for more than two different affiliations.

In case there is only one affiliation, that affiliation is given by `\paperaffiliation` (which should be set once and for all to your own affiliation), and you use the `\email` command to specify the list of email addresses to the authors.

In case of two affiliations, you call the commands `\affilblockleft`,`\affilblockrig` `\emailleft`, and `\emailright` with the appropriate arguments. Note that one of the two affiliation block arguments should simply be `\paperaffiliation`.

Additional information about the publication is given in after `\item` commands inside the `paperinfo` environment. In addition to the items given, the environment automatically starts with one item displaying the author information (without any marks related to affiliation blocks). Three commands are defined by *rtthesis* to simplify consistent formatting of additional information.

- `\paperedited`*`bib-key`* — For ordinary publications. The extent to which the publication has been edited should be state clearly. The bibliography entry will be formatted using the technique described in avsnitt B.13.

- `\paperprelver`*`ISY-report-number`* — For publications for which there is only a preliminary version available. The preliminary version should be published as a technical report at the department, and as no bibliography keys are involved, the technical report will not be listed in any the bibliography.

- `\papertechrep`*`ISY-report-number`* — For publications that are not yet even preliminary versions of something. These too should be published as technical reports at the department, and will not appear in the bibliography.

At this point the chapter title page will be finished. The next step is to make a nice title and abstract for your publication on the following odd page. Use `\maketitle` or `\maketitletwoaffil` depending on whether you set up one or two affiliation blocks. Then put the publication abstract inside the `abstract` environment.

After this point, you should just be able to include the source of your publication, with `\section` as the topmost sectioning command (since the publication itself is a chapter of your thesis).

Finally, you must decide where your references should go. Should there be one global bibliography for the whole thesis, or should there be one bibliography for each publication. This is the topic of the next section.

B.21.2 Compilation theses and bibliographies

If you are fine with having just one global bibliography for the whole thesis, everything should work out of the box. Hence, this section will try to describe how to do in order to get one bibliography for the background part of your thesis, and one for each publication.

The *rtthesis* class only supports this by relying on the bibunits package. Due to package loading order issues, it should always be loaded by passing `usebibunits` to *rtthesis*. Note that some of the bibunits commands appears to be incompatible with bibliography back references, so you need to pass the `nobackref` to *rtthesis* if you plan to use the bibunits features.

Remark B.10. There is a very interesting package called biblatex which is currently in beta version. Hopefully, it will let us drop the messy packages bibunits and backref. You are invited to try this package, and if you find it to work satisfactory it should probably be incorporated in *rtthesis*. Future maintainers of *rtthesis* are strongly encouraged to find out what biblatex can do for us!

Use the command `\defaultbibliography` to specify the bibliography files to use for all of the per-publication bibliographies, and use `\defaultbibliographystyle` to select the bibliography style, see the bibunits documentation for details.

To get an individual bibliography for a publication, you should just have to include that chapter in a `bibunit` environment, and call `\putbib` where you want the bibliography to appear. Here, the `\putbib` command will be redefined by *rtthesis* in order to make the bibliography appear in the table of contents.

A bibliography for references that appear in the background part of your thesis are produced as usual with the `\bibliography` command. (It might be good to know that *rtthesis* will automatically issue the `\nobibliography*` command in order to make the bibentry package work as you would expect.)

B.22 Master's theses

The liuthesis class by Gustaf Hendeby was developed for the production of master's theses at Linköping University. The class knows how to create the special pages required by several departments, and in the summer of 2011 this capability was merged into *rtthesis*. This makes it convenient to produce a master's thesis at Linköping University using *rtthesis* instead of liuthesis, allowing a wider audience to benefit from the more active development of *rtthesis*.[3]

This section describes how to use *rtthesis* to produce a master's thesis. To begin, pass *msc* as the value for the key *type* in the call to `\setupThesis`, and select your department using the key *department*. More details are given below, and the reader is encouraged to study the bundled example in order to get a better overall picture.

B.22.1 Master's thesis setup

In addition to the pieces of information given to `\setupThesis` for licentiate's and PhD theses (see tabell B.2), there are some that only apply to master's theses. These are listed in tabell B.14.

[3] The LaTeX class files tend to be maintained by PhD students, and PhD students have a tendency to be more interested in maintaining the class files for writing licentiate's and PhD theses than class files for master's theses.

Table B.14: *`\setupThesis` key-value pairs for master's theses, in addition to those listed in tabell B.2. Note that values that include white space are surrounded by braces.*

Key	=	Example value	Comment
swetitle	=	{Svensk titel}	Title in Swedish
swesubtitle	=	{Bra grejer}	Optional Swedish subtitle
month	=	4	
day	=	9	
subject	=	reglerteknik	
site	=	{Bosses AB i Linkan}	
division	=	{Avdelningenrt...}	
department	=	isy	See tabell B.15
examiner	=	{Lena Lärare...}	Details given below
supervisor	=	{Doktorand Si}	Details given below
keywords	=	{this, that}	Appears on library page
isrn	=	LiTH-ISY-EX...	See below
url	=	{http://...}	Thesis download URL, see below

The value for the key *department* must be one of the special values listed in tabell B.15. This setting controls both the department name and address, as well as how the special pages of the thesis are formatted. Please help the *rtthesis* maintainer to keep the special pages for your department up to date.

In the values for the keys *examiner* and *supervisor*, multiple persons should be separated using `\AND`, and the affiliation of a person should appear after `\AT`, like this:

```
supervisor={Doktorand Si \AT \textsc{isy}, Linköpings universitet
       \AND Ingenjör Så \AT Företaget},
```

The ISRN[4] should be something like

```
isrn=LITH-ISY-EX-{}-YY/NNNN-{}-SE
```

but the format varies between different departments. Note that if the report identifier contains two or three consecutive dashes, they have to be separated by empty braces in the input to prevent LaTeX from interpreting them as one character. The thesis download URL should be something like

```
url={http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-XXXXX}
```

The exact details regarding the report number and URL will be given to you by the librarian when you register your thesis.

[4]The ISO standard for ISRN was withdrawn in 2007, but the report numbering system is still in use at Linköping University.

Table B.15: *Recognized values for the key* department *in tabell B.14.*

department	**Department of...**	**Updated**
ida	Computer and Information Science	Not after 2008-08-01
ifm	Physics, Chemistry and Biology	2011-07-03
iei	Management and Engineering	*Out of date!*
isy	Electrical Engineering	2011-07-03
itn	Science and Technology	2011-07-03
mai	Mathematics	2011-07-03

B.22.2 Special pages

The requirements on a master's thesis include that certain information go on the front page and title page of the thesis. Further, a library page for cataloging purposes is required at the beginning of the thesis, and a page with copyright information is required at the end. The copyright page is automatically added at the end. The other special pages can be produced using the macros `\makeFrontPage`, `\maketitle` (as usual), and `\makeLibraryPage`. These macros are meant to be invoked more or less immediately after `\begin{document}`, see the bundled example for details. Note that in the printed report, the front page should be replaced by the cover, and the library page is *probably* meant to be on a loose piece of paper inserted between the cover and the title page.

There is no magic that puts the correct abstract on the library page, but the abstract must be given as an argument to `\makeLibraryPage`. To make sure that this is exactly the same as the abstract in the thesis, it is recommended that you write the abstract text without any surrounding `abstract` environment in a separate file, say *svensk-sammanfattning.tex*. Then you can use this file twice, like this:

```
\makeLibraryPage{\input{svensk-sammanfattning}}

\begin{abstract}[swedish]
  \input{svensk-sammanfattning}
\end{abstract}
```

(The bundled example uses this technique.)

B.22.3 Choice of language

If your main report language will be Swedish, put

```
\selectlanguage{swedish}
```

right after

```
\begin{document}
```

Also make sure to provide the thesis title (and possibly subtitle) in Swedish via the keys *swetitle* and *swesubtitle* to `\setupThesis`. You may then omit writing an abstract in English.

If your main report language will be English you don't need to change the default choice of language. However, you must provide a thesis title both in English and Swedish, and the thesis should contain abstracts in both English and Swedish.

B.23 Compiling the document

Using all the current features of *rtthesis*, the following sequence of steps is usually sufficient to compile your document. Let us assume your main file is named *main.tex*.

- First run

  ```
  pdflatex main
  ```

  to scan your document for references, labels, and index items.

- Then run

  ```
  bibtex main
  ```

  to extract relevant references from your bibliography file(s). If you are using the bibunits package, you also have to process some additional files;

  ```
  bibtex bu1; bibtex bu2; ...; bibtex bun
  ```

- If you have an index in your document, run

  ```
  makeindex main
  ```

  to format it.

- Then run

  ```
  pdflatex main
  ```

  to insert references in the typeset document. This will typically move things around, and your page references will be invalidated.

- Hopefully, it is enough to run

  ```
  pdflatex main
  ```

  once more now to get the page references right. You will get a warning if you need to repeat this step.

In addition to the steps above, certain auxiliary files must be deleted when certain features of the class are turned on or off. In particular, turning hyperlinks on or off requires the following.

```
rm main.aux main.toc main.ind
```

Table B.16: *Files with logotype graphics used by* rtthesis. *Use the command* `kpsewhich` *to find where the files are located!*

Filename	Use
LinkUniv_sigill_sv.pdf	For the cover and the first page in PhD theses.
LiTH_staende_eng_sv.pdf	For the cover of both licentiate's and PhD theses.
rtlogo_tall.pdf	For the first page in licentiate's theses.

B.24 Generating a thesis cover and the "spikblad"

A thesis cover can be created by making a file that contains the `\makecover` command. For example, given that *mythesis.sty* invokes the `\setupThesis` command with the necessary information (see tabell B.2), a PhD thesis cover can be made as follows.

```
\documentclass[utf8,phd]{rtthesis}
\usepackage{mythesis}


\makecover
```

Note that while all licentiate's theses should have the same cover, there is no standard (but many rules set by the university!) for the PhD theses. The `\makecover` command gives a "classic" cover that quite a few people have used over the years. This cover might also be useful as a means to compile the information needed when LiU-Tryck (or some other printing company) designs a more artistic cover.

For a dissertation, there should always be a "spikblad" (literally, *nailing sheet*). Such an information sheet can be generated easily if the English abstract is put in a separate file. In this case, the same abstract can be included both in the thesis and in a separate file that defines the "spikblad". For a licentiate's thesis presentation, a similar information sheet should be produced. The monograph example demonstrates how to created these, see the files *spikblad.tex* (for dissertations) and *licinfo.tex* (for licentiate's thesis presentations).

B.25 Required logotypes (not included with *rtthesis*)

Tabell B.16 lists files with logotype graphics that are needed by *rtthesis*. They are not part of the *rtthesis* bundle since they are used in many other contexts as well. Users at the Division of Automatic Control should have access to these files via the group's common texmf tree, but in order to be able to work at home you will have to make sure one way or another that the files are installed.

Beware that the university changes logos quite often. Make sure that there are no new versions of the logos you use. If the logos are old, please, let the *rtthesis* maintainer know so that the files get updated at the central location.

B.26 Compatibility with standard packages

Incompatibilities between different packages is a problem that quickly becomes quite an issue when the list of packages used in a document grows beyond just a few. It may sound strange, but it is because of compatibility problems that *rtthesis* includes a rather long list of packages for you. The reason is that this allows knowledge about package loading order requirements and various workarounds, to be encoded in the class file.

No list of packages included by *rtthesis* will be presented here, but you should check the class file directly to be sure that you always get the correct answer to whether a package is included or not (or you can just read the compilation output).

Packages with no known compatibility issues will generally not be included by *rtthesis* unless needed by the class itself. The following list contains some examples of useful packages that are not included by *rtthesis*. They *should* be compatible with *rtthesis*. Please let the *rtthesis* maintainer know if any of these are no longer compatible, or if you have suggestions for other packages that should be mentioned here.

- nextpage — page break control
- algorithm — code listings
- listings — code listings
- SIunits — physical dimensions
- pmat — partitioned matrices
- bm — bold math
- footmisc — extras for footnotes
- dcolumn — decimal point alignment in tables (the already included array can also do this)
- lettrine — start chapter with fancy letter
- supertabular — multi-page tables
- longtable — multi-page tables
- multirow — tabular entries occupying more than one row

Bibliography

Apple Inc. Truetype™reference manual, 2015. URL https://developer.apple.com/fonts/TrueType-Reference-Manual/. Cited on page 3.

Xiao-Diao Chen, Yin Zhou, Zhenyu Shu, Hua Su, and J.-C. Paul. Improved algebraic algorithm on point projection for beziércurves. In *Computer and Computational Sciences, 2007. IMSCCS 2007. Second International Multi-Symposiums on*, pages 158–163, Aug 2007. doi: 10.1109/IMSCCS.2007.17. Cited on page 5.

Olivier Cuisenaire and Benoît Macq. Fast euclidean distance transformation by propagation using multiple neighborhoods. *Computer Vision and Image Understanding*, 76(2):163–172, 1999. Cited on page 9.

Per-Erik Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14(3):227 – 248, 1980. ISSN 0146-664X. doi: http://dx.doi.org/10.1016/0146-664X(80)90054-4. URL http://www.sciencedirect.com/science/article/pii/0146664X80900544. Cited on pages 7 and 8.

Hinnik Eggers. Two fast euclidean distance transformations in z 2 based on sufficient propagation. *Computer Vision and Image Understanding*, 69(1):106–116, 1998. Cited on page 9.

Ricardo Fabbri, Luciano Da F. Costa, Julio C. Torelli, and Odemir M. Bruno. 2d euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv.*, 40(1):2:1–2:44, February 2008. ISSN 0360-0300. doi: 10.1145/1322432.1322434. URL http://doi.acm.org/10.1145/1322432.1322434. Cited on pages 8 and 9.

Kaspar Fischer. Piecewise linear approximation of bezier curves. In *HTTP://HCKLBRRFNN. WORDPRESS. COM/2012/08/20/PIECEWISE-LINEAR-APPROXIMATION-OF-BEZIER-CURVES/*. Citeseer, 2000. Cited on page 5.

FreeType. Freetype outlines, 2014. URL http://www.freetype.org/freetype2/docs/glyphs/glyphs-6.html. Cited on page 3.

Chris Green. Improved alpha-tested magnification for vector textures and special effects. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, pages 9–18, New York, NY, USA, 2007. ACM. ISBN 978-1-4503-1823-5. doi: 10.1145/1281500.1281665. URL http://doi.acm.org/10.1145/1281500.1281665. Cited on page 9.

Stefan Gustavson and Robin Strand. Anti-aliased euclidean distance transform. *Pattern Recognition Letters*, 32(2):252 – 257, 2011. ISSN 0167-8655. doi: http://dx.doi.org/10.1016/j.patrec.2010.08.010. URL http://www.sciencedirect.com/science/article/pii/S0167865510002953. Cited on page 9.

M. Jones, J.A. Baerentzen, and M. Sramek. 3d distance fields: a survey of techniques and applications. *Visualization and Computer Graphics, IEEE Transactions on*, 12(4):581–599, July 2006. ISSN 1077-2626. doi: 10.1109/TVCG.2006.56. Cited on page 4.

Hassan K. Khalil. *Nonlinear systems*. Prentice Hall, Inc., third edition, 2002. Cited on page 30.

Roberto A Lotufo and Francisco A Zampirolli. Fast multidimensional parallel euclidean distance transform based on mathematical morphology. In *Computer Graphics and Image Processing, 2001 Proceedings of XIV Brazilian Symposium on*, pages 100–105. IEEE, 2001. Cited on page 9.

Calvin R Maurer Jr, Vijay Raghavan, and Rensheng Qi. A linear time algorithm for computing the euclidean distance transform in arbitrary dimensions. In *Information Processing in Medical Imaging*, pages 358–364. Springer, 2001. Cited on page 9.

Arnold Meijster, Jos BTM Roerdink, and Wim H Hesselink. A general algorithm for computing distance transforms in linear time. In *Mathematical Morphology and its applications to image and signal processing*, pages 331–340. Springer, 2000. Cited on page 9.

Microsoft Corporation. Opentype specification, 2015. URL https://www.microsoft.com/typography/otspec/default.htm. Cited on page 3.

Lapo Filippo Mori. Tables in LaTeX 2_ε: Packages and methods. *The PracTeX Journal*, 2007(1), 2007. URL http://tug.org/pracjourn/2007-1/mori. Cited on pages 26 and 32.

Thomas W Phinney. Truetype, postscript type 1, & opentype: What's the difference, 2001. Cited on page 5.

Ingemar Ragnemalm. *The Euclidean Distance Transform*. Linköping Studies in Science and Technology, first edition, 1993. ISBN 91-7871-083-9. Cited on pages 4 and 8.

Ingemar Ragnemalm. *Polygons feel no pain*. First edition, 2008. Cited on page 5.

Azriel Rosenfeld and John L Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM (JACM)*, 13(4):471–494, 1966. Cited on page 4.

Wilson J. Rugh. *Linear system theory*. Prentice-Hall, Inc., second edition, 1996. Cited on page 30.

Marek Ryćko and Bugosław Jackowski. TEX from \indent to \par. *TUGboat*, 14(3), 1993. URL `http://www.tug.org/TUGboat/Articles/tb14-3/tb40rycko.pdf`. Cited on pages 30 and 31.

Toyofumi Saito and Jun-Ichiro Toriwaki. New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications. *Pattern recognition*, 27(11):1551–1565, 1994. Cited on page 9.

Dave Shreiner, Bill The Khronos OpenGL ARB Working Group, et al. *OpenGL programming guide: the official guide to learning OpenGL, versions 3.0 and 3.1*. Pearson Education, 2009. Cited on page 5.

Index