graphics/Logo-schwarz.pdf

Auswirkungen des zeitlichen Kontexts auf die Robustheit in der UAV-gestützten Bildverarbeitung

Optionaler Untertitel der Arbeit

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Moritz Anton Zideck
Matrikelnummer 12217036

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Senior Lecturer Dipl.-Ing. Dr.techn. Sebastian Zambanini
Mitwirkung: Dipl. Inf Marvin Burges

Wien, 1. Jänner 2001

	Moritz Anton Zideck	Sebastian Zambanini

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

graphics/Logo-schwarz.pdf

Impact of Temporal Context on Robustness in UAV-based Imagery

Optional Subtitle of the Thesis

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Moritz Anton Zideck
Registration Number 12217036

to the Faculty of Informatics

at the TU Wien

Advisor: Senior Lecturer Dipl.-Ing. Dr.techn. Sebastian Zambanini
Assistance: Dipl. Inf Marvin Burges

Vienna, January 1, 2001

Moritz Anton Zideck

Sebastian Zambanini

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Moritz Anton Zideck

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, habe ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT-Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 1. Jänner 2001

Moritz Anton Zideck

Danksagung

Ihr Text hier.

Acknowledgements

Enter your text here.

Kurzfassung

Ihr Text hier.

Abstract

Enter your text here.

Contents

CHAPTER 1

Introduction

Video object detection is a relatively new field in computer vision that aims to leverage the temporal context of videos in order to improve detection performance compared to image-based object detection. Temporal context refers to information that can be extracted from the time domain of a video, such as object motion, object trajectories, and changes in object appearance over time. By exploiting this additional information, video object detection models can achieve higher accuracy, improved robustness, and greater temporal consistency when detecting objects across frames. This is especially important in scenarios where objects may be occluded, blurred, or undergo significant appearance changes over time. A prominent example is UAV-based imagery, where the camera is constantly moving, objects are often small, and visual conditions can change rapidly, making reliable object detection particularly challenging.

To date, most evaluations of object detection models—both image-based and video-based—focus primarily on accuracy measured on clean and unperturbed data. However, in real-world deployments, and especially in UAV-based applications, visual data is frequently affected by a wide range of perturbations, including sensor noise, compression artifacts, motion blur, illumination changes, and variations in viewpoint and scale. These perturbations can substantially degrade detection performance. Although several works have investigated the robustness of image-based object detectors to such corruptions, only limited attention has been given to the robustness of video object detection models, and virtually no systematic studies exist for UAV-based video data in particular.

In this thesis, the focus is therefore placed on the robustness of video object detection models under common perturbations in UAV-based imagery. Robustness is defined as the ability of a model to maintain its detection performance when exposed to adverse conditions such as lighting changes, weather effects, motion blur, occlusions, and variations in object appearance and scale. A model may achieve high accuracy on clean data, yet still be unreliable in practice if its performance deteriorates significantly under realistic perturbations. For safety-critical and autonomous UAV applications, such robustness is essential.

The central research question of this thesis is how the incorporation of temporal context in video object detection models affects their robustness to common perturbations in UAV-based imagery. In particular, the impact of the number of reference frames used to construct the temporal context is analyzed. To address this question, a comprehensive evaluation of state-of-the-art video object detection models is conducted on a benchmark dataset designed for UAV scenarios. In addition, a novel robustness evaluation metric is proposed, which quantifies the performance degradation of a model under different perturbations relative to its performance on clean data.

CHAPTER 2

Additional Chapter

Enter your text here.

CHAPTER 3

Method

3.1 Models

For this thesis two video detector models are chosen, which represent different approaches to leverage temporal context in different ways. On the one hand there is TransVOD [?], which is a transformer based model that uses attention mechanisms to aggregate temporal information from multiple frames. On the other hand YOLOV [?], which is a one-stage detector that extends the popular YOLO architecture to video data by incorporating temporal feature fusion techniques. For both models the Swin base backbone [LLC+21] has been chosen, to ensure a fair compairson as well as petrain weights being available for both models.

Together these models provide a good basis for evaluation, as they represent different design philosophies and represent pro and cons in terms of temporal context utilization, computational efficiency and detection accuracy.

3.1.1 Transvod

Transvod [?] is a end to end video object detection model base on DETR [CMS+20]. End to end mean that no hand crafted features as well as no post processing is needed, everthing is learned by the model itself. The models fist version was proposed in 2021 as one of the first transformer based video object detection models to streamline the detection pipeline and remove the need for hand crafted features. By encoding not only spatial but also temporal information in their attention mechanism, the model shows strong performance on various video object detection benchmarks. In the most well known video object detection benchmark, ImageNet VID [RDS+15] outperforms its single frame baseline by 3.6 mAP(%) achieving 80.7 mAP(%) on the validation set. One year later an improved version of TransVOD was proposed, called TransVOD++ [?], which builds upon the original TransVOD architecture and introduces several enhancements to further improve detection performance. The main goal of TransVOD++ is to address the heavy computation costs as well as increase the detection accuracy of its predecessor. Next to architectural improvements, which i will describe in the following, a new backbone namely Swin-Base [LLC^{+}21] instead of ResNet-101 [HZRS16] was used to further boost performance. With these improvements TransVOD++ was the first model to achieve over 90 mAP(%) on the ImageNet VID validation set, reaching 90.0 mAP(%). TransVod short summary: ...

Model design

3.1.2 YOLOV

Base on the popular YOLO [RDGF16] architecture, to be more precise YOLOX [GLW^{+}21][redacted] which are one-stage detectors known for their speed and efficiency, YOLOV [**?**] extends this architecture to video data by incorporating temporal feature fusion techniques. The paper that was published in 2023, was able to surpass previous state of the art video object detection models on the ImageNet VID [RDS^{+}15] benchmark with a mAP(%) of 85.5 on the validation set, while being still near real time capable with 22.7 FPS on a Nvidia TITAN RTX GPU. YOLOV achieves this by introducing a temporal feature fusion module that aggregates features from multiple frames, allowing the model to leverage temporal context effectively. Futhermore like TransVOD a improved version of YOLOV was proposed called YOLOV++ [SZG24], which further enhances the temporal feature fusion mechanism and introduces additional optimizations to improve detection accuracy and efficiency. The now improved YOLOV++ was able to achieve a new record mAP(%) of 93.2 on the ImageNet VID validation set, while still being over 30 FPS on a Nvidia RTX 3090. However for this a special backbone names FocalNet-Large [YLDG22] is used. Using the same Swin-Base [LLC^{+}21] backbone as for TransVOD++, YOLOV++ achieves a mAP(%) of 90.7 on the ImageNet VID validation set, which is still significantly higher than TransVOD++ with 90.0 mAP(%).

Model design

3.2 Visdrone Dataset

The Visdrone dataset [ZWD^{+}21] is a large-scale dataset for different detection scenarios based on drone based imagery. The images and videos were captured by various drone platforms in different urban and suburban areas of 14 different cities across China. The objects are entitys of public street scenes, e.g., pedestrians, vehicles, bicycles, etc. All together there are 10 different object categories. It is curated by the *AISKYEYE* research group from the *Tianjin University in China.* For this thesis VISDRONE2019-VID is used, to leverage the temporal context of the videos for the object detection task. All together the dataset contains 79 sequences with 33,366 frames, which are split 56 videos with 24,198 frames for training, 7 videos with 2,846 frames for validation and 16 videos with 6,322 frames for testing. The dataset is chosen because of its large size and the challenging scenarios, e.g., different weather conditions, various altitudes and camera angles as well as high density of objects in the images. Object sizes vary significantly, ranging from very small objects with only a few pixels to large objects covering a significant portion of the image. This large difference in object sizes makes it also suitable to evaluate the performance of detection models under different perturbations and across different scales.

graphics/visdrone_example_1.jpg

(a) Example 1

graphics/visdrone_example_2.jpg

(b) Example 2

graphics/visdrone_example_3.jpg

(c) Example 3

Figure 3.1: Example images from the Visdrone dataset.

CHAPTER 4

Perurbation

A signficant part of this thesis to evaluate the robustness of video object detection models under common perturbations in UAV-based imagery. In this sentence two key concepts must be defined: what robustness means in the context of an object detection model, and which perturbations commonly occur in UAV-based imagery.

4.0.1 Definition of Robustness

A widely used definition of robustness was proposed by Hendrycks and Dietterich [HD19], who define robustness as a model's ability to maintain predictive performance under distribution shifts caused by common, naturally occurring image corruptions and perturbations. In the context of object detection, this means that a robust model should be able to accurately detect and localize objects even when the input images are affected by various types of noise, distortions, or other adverse conditions. In the context of this thesis, the main metric to quantify robustness is the relative performance degradation of a model mean avarage precision (mAP) under different perturbations compared to its performance on clean data.

4.0.2 Common Perturbations in UAV-based Imagery

The paper by Hendrycks and Dietterich [HD19] introduced a benchmark suite called ImageNet-C, which consists of 15 different types of common image corruptions applied to the ImageNet dataset. For further insight a paper named *Benchmarking the Robustness of UAV Tracking Against Common Corruptions* [LFH$^+$24] which was published in 2024, is used to identify perturbations that commonly occur in UAV-based imagery. Based on these works, the following perturbations are considered in this thesis:

- Gaussian Noise: Random noise following a Gaussian distribution is added to the image pixels, simulating sensor noise.

- Motion Blur: Simulates the effect of camera or object motion during exposure, resulting in blurred images.
- Defocus Blur: Simulates the effect of an out-of-focus lens, resulting in blurred images.
- Brightness Changes: Adjusts the overall brightness of the image, simulating different lighting conditions.
- Contrast Changes: Adjusts the contrast of the image, affecting the distinction between light and dark areas.
- Jpeg Compression: Simulates artifacts introduced by JPEG compression at various quality levels.

The simulation of weather conditions such as fog, rain, and snow is not considered in this thesis, as these perturbations require more complex rendering techniques. For each perturbation type, multiple severity levels are defined to assess robustness across a range of adverse conditions; in this thesis, three levels are used: low, medium, and high.

4.0.3 Implementation

To apply the defined perturbations to the Visdrone dataset, a custom data augmentation pipeline is implemented into to the models data loader. Based on evaluation input parameters, the data loader applies the specified perturbation with the desired severity level to each frame before it is fed into the model for inference. For more in depth evaluation a probability parameter is added, which defines the likeliness each perturbation being applied to a frame.

Gaussian noise. We add i.i.d. Gaussian noise:

$$\tilde{I} = \mathrm{clip}\big(I + N,\ 0,\ 255\big), \qquad N_{h,w,c} \sim \mathcal{N}\Big(0,\ (\sigma \cdot 255)^2\Big), \tag{4.1}$$

where σ is the noise standard deviation.

Defocus blur. We approximate defocus blur by convolving the image with a normalized disk (pillbox) kernel:

$$\tilde{I} = I * K_{\text{disk}}, \qquad K_{\text{disk}}(u,v) = \frac{1}{Z}\mathbb{1}\Big(u^2 + v^2 \leq r^2\Big), \tag{4.2}$$

where K_{disk} is a $k \times k$ kernel, $r = \lfloor k/2 \rfloor$, $Z = \sum_{u,v} \mathbb{1}(\cdot)$, and $*$ denotes 2D convolution.

Motion blur. We simulate linear motion blur by convolving with a sparse line kernel of size $k \times k$ oriented by an angle θ (in degrees, default $\theta = 0$). The kernel is constructed by placing ones on the discrete line

$$v = \tan(\theta)\, u, \qquad u \in \big[-\lfloor k/2 \rfloor\,,\ \lfloor k/2 \rfloor\big]\,, \tag{4.3}$$

rasterized onto the kernel grid and normalized to sum to one, then $\tilde{I} = I * K_{\text{motion}}$.

Brightness change. We apply a global multiplicative gain:

$$\tilde{I} = \mathrm{clip}(\alpha I,\ 0,\ 255), \tag{4.4}$$

with $\alpha > 0$.

Contrast change. We scale deviations from the per-channel mean:

$$\mu_c = \frac{1}{HW}\sum_{h,w} I_{h,w,c}, \qquad \tilde{I}_{h,w,c} = \mathrm{clip}((I_{h,w,c} - \mu_c)\alpha + \mu_c,\ 0,\ 255), \tag{4.5}$$

with contrast factor α.

Pixelation. We downsample and upsample the image using a block factor p (default $p = 8$). Specifically, we resize I to $(\lfloor W/p \rfloor, \lfloor H/p \rfloor)$ using bilinear interpolation, then resize back to (W, H) using nearest-neighbor interpolation:

$$\tilde{I} = \mathrm{NN}(\mathrm{BL}(I; \lfloor W/p \rfloor, \lfloor H/p \rfloor);\ W, H), \tag{4.6}$$

where BL denotes bilinear resize and NN denotes nearest-neighbor resize.

JPEG compression. We simulate compression artifacts by encoding and decoding the image using JPEG with quality parameter q:

$$\tilde{I} = \mathrm{JPEGdecode}(\mathrm{JPEGencode}(I; q)). \tag{4.7}$$

The specific severity levels are defined as follows:

Table 4.1: Perturbation presets and severity levels used in robustness evaluation.

Perturbation	**Low**	**Medium**	**High**
Gaussian noise	$\sigma = 0.01$	$\sigma = 0.05$	$\sigma = 0.10$
Defocus blur	$k = 3$	$k = 7$	$k = 11$
Motion blur	$k = 3,\ \theta = 0°$	$k = 7,\ \theta = 0°$	$k = 15,\ \theta = 0°$
Brightness change	$\alpha = 1.10$	$\alpha = 1.25$	$\alpha = 1.45$
Contrast change	$\alpha = 1.10$	$\alpha = 1.25$	$\alpha = 1.45$
Pixelation	$p = 2$	$p = 4$	$p = 6$
JPEG compression	$q = 85$	$q = 55$	$q = 25$

With this setup, all together 18 different perturbation configurations (6 perturbation types × 3 severity levels) can be evaluated to assess the robustness of video object detection models in UAV-based imagery. To give an impression of the applied perturbations, example images for each perturbation type and severity level are shown in Figure **??**.

CHAPTER 5

Evaluation Metric

CHAPTER 6

Experiments

CHAPTER 7

Results

CHAPTER 8

Conclusion

CHAPTER 9

Introduction to LaTeX

Since LaTeX is widely used in academia and industry, there exists a plethora of freely accessible introductions to the language. Reading through the guide at `https://en.wikibooks.org/wiki/LaTeX` serves as a comprehensive overview of most of the functionality and is highly recommended before starting with a thesis in LaTeX.

9.1 Installation

A full LaTeX distribution consists not only of the binaries that convert the source files to the typeset documents but also of a wide range of packages and their documentation. Depending on the operating system, different implementations are available as shown in Table 9.1. **Due to the large number of packages that are in everyday use and due to their high interdependence, it is paramount to keep the installed distribution up to date.** Otherwise, obscure errors and tedious debugging ensue.

9.2 Editors

A multitude of TeX editors are available differing in their editing models, their supported operating systems, and their feature sets. A comprehensive overview of editors can be

Distribution	Unix	Windows	MacOS
TeX Live	**yes**	yes	(yes)
MacTeX	no	no	**yes**
MikTeX	(yes)	**yes**	yes

Table 9.1: TeX/LaTeX distributions for different operating systems. Recommended choice is in **bold**.

	Description
1	Scan for refs, toc/lof/lot/loa items and cites
2	Build the bibliography
3	Link refs and build the toc/lof/lot/loa
4	Link the bibliography
5	Build the glossary
6	Build the acronyms
7	Build the index
8	Link the glossary, acronyms, and the index
9	Link the bookmarks
	Command
1	`pdflatex.exe  example`
2	`bibtex.exe    example`
3	`pdflatex.exe  example`
4	`pdflatex.exe  example`

5	`makeindex.exe -t example.glg -s example.ist`
	`-o example.gls example.glo`

6	`makeindex.exe -t example.alg -s example.ist`
	`-o example.acr example.acn`

7	`makeindex.exe -t example.ilg -o example.ind example.idx`
8	`pdflatex.exe  example`
9	`pdflatex.exe  example`

Table 9.2: Compilation steps for this document. The following abbreviations were used: table of contents (toc), list of figures (lof), list of tables (lot), list of algorithms (loa).

found on the Wikipedia page `https://en.wikipedia.org/wiki/Comparison_of_TeX_editors`. TeXstudio (`http://texstudio.sourceforge.net/`) is recommended. Most editors support synchronization of the generated document and the LaTeX source by `Ctrl`-clicking either on the source document or the generated document.

9.3 Compilation

Modern editors usually provide the compilation programs to generate Portable Document Format (PDF) documents and for most LaTeX source files, this is sufficient. More advanced LaTeX functionality, such as glossaries and bibliographies, needs additional compilation steps, however. It is also possible that errors in the compilation process invalidate intermediate files and force subsequent compilation runs to fail. It is advisable to delete intermediate files (`.aux`, `.bbl`, etc.), if errors occur and persist. All files that are not generated by the user are automatically regenerated. To compile the current document, the steps as shown in Table 9.2 have to be taken.

9.4 Basic Functionality

In this section, various examples are given of the fundamental building blocks used in a thesis. Many LaTeX commands have a rich set of options that can be supplied as optional arguments. The documentation of each command should be consulted to get an impression of the full spectrum of its functionality.

9.4.1 Floats

Two main categories of page elements can be differentiated in the usual LaTeX workflow: *(i)* the main stream of text and *(ii)* floating containers that are positioned at convenient positions throughout the document. In most cases, tables, plots, and images are put into such containers since they are usually positioned at the top or bottom of pages. These are realized by the two environments `figure` and `table`, which also provide functionality for cross-referencing (see Table 9.3 and Figure 9.1) and the generation of corresponding entries in the list of figures and the list of tables. Note that these environments solely act as containers and can be assigned arbitrary content.

9.4.2 Tables

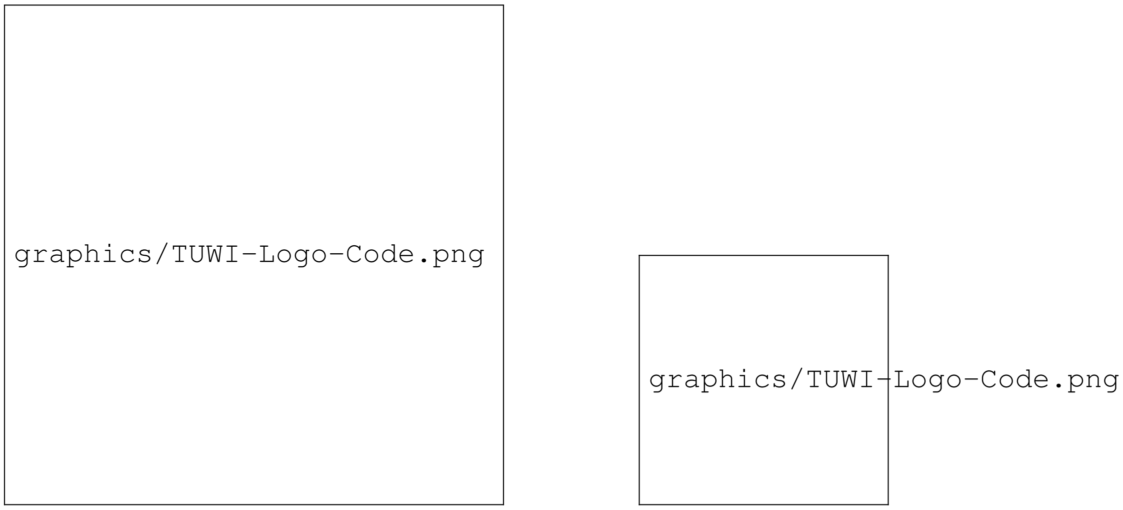
A table in LaTeX is created by using a `tabular` environment or any of its extensions, e.g., `tabularx`. The commands `\multirow` and `\multicolumn` allow table elements to span multiple rows and columns.

Position		
Group	Abbrev	Name
Goalkeeper	GK	Paul Robinson
Defenders	LB	Lucus Radebe
	DC	Michael Duburry
	DC	Dominic Matteo
	RB	Didier Domi
Midfielders	MC	David Batty
	MC	Eirik Bakke
	MC	Jody Morris
Forward	FW	Jamie McMaster
Strikers	ST	Alan Smith
	ST	Mark Viduka

Table 9.3: Adapted example from the LaTeX guide at `https://en.wikibooks.org/wiki/LaTeX/Tables`. This example uses rules specific to the `booktabs` package and employs the multi-row functionality of the `multirow` package.

9.4.3 Images

An image is added to a document via the `\includegraphics` command as shown in Figure 9.1. The `\subcaption` command can be used to reference subfigures, such as Figure 9.1a and 9.1b.



(a) The TU Wien Informatics logo at text width.

(b) The TU Wien Informatics logo at half the text width.

Figure 9.1: The header logo at different sizes.

9.4.4 Mathematical Expressions

One of the original motivations for creating the TeX system was the need for mathematical typesetting. To this day, LaTeX is the preferred system to write math-heavy documents and a wide variety of functions aids the author in this task. A mathematical expression can be inserted inline as $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$ outside of the text stream as

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

or as a numbered equation with

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}. \tag{9.1}$$

9.4.5 Pseudo Code

The presentation of algorithms can be achieved with various packages; the most popular are `algorithmic`, `algorithm2e`, `algorithmicx`, or `algpseudocode`. An overview is given at `https://tex.stackexchange.com/questions/229355`. An example of the use of the `alogrithm2e` package is given with Algorithm 9.1.

Algorithm 9.1: Gauss-Seidel

```
Input: A scalar $\epsilon$, a matrix $\mathbf{A} = (a_{ij})$, a vector $\vec{b}$, and an initial vector $\vec{x}^{(0)}$
Output: $\vec{x}^{(n)}$ with $\mathbf{A}\vec{x}^{(n)} \approx \vec{b}$
for $k \leftarrow 1$ to maximum iterations do
    for $i \leftarrow 1$ to $n$ do
        $x_i^{(k)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j<i} a_{ij} x_j^{(k)} - \sum_{j>i} a_{ij} x_j^{(k-1)} \right)$;
    end
    if $|\vec{x}^{(k)} - \vec{x}^{(k-1)}| < \epsilon$ then
        break for;
    end
end
return $\vec{x}^{(k)}$;
```

9.5 Bibliography

The referencing of prior work is a fundamental requirement of academic writing and is well supported by LaTeX. The BibTeX reference management software is the most commonly used system for this purpose. Using the `\cite` command, it is possible to reference entries in a `.bib` file out of the text stream, e.g., as [Tur36]. The generation of the formatted bibliography needs a separate execution of `bibtex.exe` (see Table 9.2).

9.6 Table of Contents

The table of contents is automatically built by successive runs of the compilation, e.g., of `pdflatex.exe`. The command `\setsecnumdepth` allows the specification of the depth of the table of contents and additional entries can be added to the table of contents using `\addcontentsline`. The starred versions of the sectioning commands, i.e., `\chapter*`, `\section*`, etc., remove the corresponding entry from the table of contents.

9.7 Acronyms / Glossary / Index

The list of acronyms, the glossary, and the index need to be built with a separate execution of `makeindex` (see Table 9.2). Acronyms have to be specified with `\newacronym` while glossary entries use `\newglossaryentry`. Both are then used in the document content with one of the variants of `\gls`, such as `\Gls`, `\glspl`, or `\Glspl`. Index items are simply generated by placing `\index{`⟨*entry*⟩`}` next to all the words that correspond to the index entry ⟨*entry*⟩. Note that many enhancements exist for these functionalities and the documentation of the `makeindex` and the `glossaries` packages should be consulted.

9.8 Tips

Since TeX and its successors do not employ a What You See Is What You Get (WYSIWYG) editing scheme, several guidelines improve the readability of the source content:

- Each sentence in the source text should start with a new line. This helps not only the user navigate through the text but also enables revision control systems (e.g. Subversion (SVN), Git) to show the exact changes authored by different users. Paragraphs are separated by one (or more) empty lines.
- Environments, which are defined by a matching pair of `\begin{name}` and `\end{name}`, can be indented by whitespace to show their hierarchical structure.
- In most cases, the explicit use of whitespace (e.g. by adding `\hspace{4em}` or `\vspace{1.5cm}`) violates typographic guidelines and rules. Explicit formatting should only be employed as a last resort and, most likely, better ways to achieve the desired layout can be found by a quick web search.
- The use of bold or italic text is generally not supported by typographic considerations and the semantically meaningful `\emph{...}` should be used.

The predominant application of the LaTeX system is the generation of PDF files via the PdfLaTeX binaries. In the current version of PdfLaTeX, it is possible that absolute file paths and user account names are embedded in the final PDF document. While this poses only a minor security issue for all documents, it is highly problematic for double-blind reviews. The process shown in Table 9.4 can be employed to strip all private information from the final PDF document.

	Command
1	Rename the PDF document `final.pdf` to `final.ps`.

2	Execute the following command:
	`ps2pdf -dPDFSETTINGS#/prepress ^`
	` -dCompatibilityLevel#1.4 ^`
	` -dAutoFilterColorImages#false ^`
	` -dAutoFilterGrayImages#false ^`
	` -dColorImageFilter#/FlateEncode ^`
	` -dGrayImageFilter#/FlateEncode ^`
	` -dMonoImageFilter#/FlateEncode ^`
	` -dDownsampleColorImages#false ^`
	` -dDownsampleGrayImages#false ^`
	` final.ps final.pdf`

On Unix-based systems, replace # with = and ^ with \.

Table 9.4: Anonymization of PDF documents.

9.9 Resources

9.9.1 Useful Links

In the following, a listing of useful web resources is given.

https://en.wikibooks.org/wiki/LaTeX An extensive wiki-based guide to LaTeX.

http://www.tex.ac.uk/faq A (huge) set of Frequently Asked Questions (FAQ) about TeX and LaTeX.

https://tex.stackexchange.com/ The definitive user forum for non-trivial LaTeX-related questions and answers.

9.9.2 Comprehensive TeX Archive Network (CTAN)

The CTAN is the official repository for all TeX-related material. It can be accessed via `https://www.ctan.org/` and hosts (among other things) a huge variety of packages that provide extended functionality for TeX and its successors. Note that most packages contain PDF documentation that can be directly accessed via CTAN.

In the following, a short, non-exhaustive list of relevant CTAN-hosted packages are given together with their relative path.

algorithm2e Functionality for writing pseudo code.

amsmath Enhanced functionality for typesetting mathematical expressions.

amssymb Provides a multitude of mathematical symbols.

booktabs Improved typesetting of tables.

enumitem Control over the layout of lists (`itemize`, `enumerate`, `description`).

fontenc Determines font encoding of the output.

glossaries Create glossaries and lists of acronyms.

graphicx Insert images into the document.

inputenc Determines encoding of the input.

l2tabu A description of bad practices when using LaTeX.

mathtools Further extension of mathematical typesetting.

memoir The document class upon which the `vutinfth` document class is based.

multirow Allows table elements to span several rows.

pgfplots Function plot drawings.

pgf/TikZ Creating graphics inside LaTeX documents.

subcaption Allows the use of subfigures and enables their referencing.

symbols/comprehensive A listing of around 5000 symbols that can be used with LaTeX.

voss-mathmode A comprehensive overview of typesetting mathematics in LaTeX.

xcolor Allows the definition and use of colors.

Overview of Generative AI Tools Used

Ihr Text hier.

Übersicht verwendeter Hilfsmittel

Enter your text here.

List of Figures

List of Tables

List of Algorithms

Index

Bibliography

[CMS⁺20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

[GLW⁺21] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[HD19] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[LFH⁺24] Xiaoqiong Liu, Yunhe Feng, Shu Hu, Xiaohui Yuan, and Heng Fan. Benchmarking the robustness of uav tracking against common corruptions. *arXiv preprint arXiv:2403.11424v1*, March 2024. [cs.CV].

[LLC⁺21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[RDS⁺15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, December 2015.

[SZG24] Yuheng Shi, Tong Zhang, and Xiaojie Guo. Practical video object detection via feature selection and aggregation. *arXiv preprint arXiv:2407.19650*, 2024.

[Tur36] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *J. of Math*, 58:345–363, 1936.

[YLDG22] Jianwei Yang, Chao Li, Xiaohang Dai, and Jianfeng Gao. Focal modulation networks. In *Advances in Neural Information Processing Systems*, 2022.

[ZWD$^+$21] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7380–7399, 2021.