TeX-to-TikZ

Alex Friedman, Charlotte Clark,
Natalie McClain, Shaw-leon Chen,
Connie Chau, Jake Backer,
N'yoma Diamond, and Rose Bohrer

This interstate paper (with work occurring in 7 different US states) is highly reviewed being edited, in part, at 27,000ft (Flight Level 270).

Acknowledgements

I would like to thank my coauthors for generously contributing their names to this paper. Without them, this paper would not have the same SHA256 hash as it has today. I hope they will all be able to go on and have illustrious careers without this critical research overshadowing their future contributions.

1 Introduction

As it stands, there are many limitations to how academic papers are typeset. In the standard typesetting tool, LaTeX, users lack granular control over the exact appearance of their text on the page. In addition, they have to worry about numerous cryptic errors (such as overfull and underfull hboxes). The use of TeX also places high barriers on collaboration due to the need to frequently create custom build sequences and install a wide variety of packages. To remedy these issues, this paper investigates the possibility to typeset research papers exclusively using TikZ. To this end, this paper outlines the TeX-to-TikZ and pdf-to-TikZ compiler which allows for backporting existing works into TikZ.

2 Background

TikZ is a commonly used library for drawing images within TeX documents [1] whose pronunciation is often debated on StackOverflow. Despite its frequent use in academic papers, little research has focused on TikZ specifically—with most describing the use of TikZ for various applications [3]. Nonetheless, several works documenting its importance, usage, and (most importantly) existence do exist [6, 2].

While new researchers to this exiting field may find our decision to replace typesetting in TeX with TikZ confusing as, after all, being a TeX library, any documents typeset with TikZ would have to use TeX, we base this work on the following postulates:

$$\mathsf{TikZCodeInPaper}(p)/\mathsf{TotalLinesOfCodeInPaper(p)} \approx 1$$

$$\lim_{p\to\infty} \frac{\sum_{P=1}^{p} \mathsf{LibrariesRequiredForTikZ(P)/p}}{\sum_{P=1}^{p} \mathsf{LibrariesNormallyRequired(P)}} = \frac{1}{\infty}$$

It thus follows that typesetting a document with TikZ results in a much less complicated toolchain than what would be otherwise required to typeset a paper—making TeX-to-TikZ clearly the more efficient option just from its time savings.

3 Implementation

To help increase the versatility of this work, I decided to make my implementation of TeX-to-TikZ convert PDF files to TeX files which only use TikZ instead of running the conversion from TeX files. This follows from the basic axiom: "I don't want to figure out how to compile your TeX document; if it works for you, just run it and send me the pdf."

As this work is currently not funded[1], I couldn't be bothered to learn the entire PDF specifications prior to the paper's deadline. Because of this, the decision was made to implement a prototype of TeX-to-TikZ [4]. This has the added benefit of preserving the full TeX-to-TikZ implementation for my thesis. As such, in this preliminary implementation, we immediately squander all potential time savings of TeX-to-TikZ through scanning a PDF pixel by pixel, and writing the TikZ code for drawing said pixel to an output file. As PDFs are vector images, and thus have an infinite resolution, our prototype scans the document at a capped resolution to ensure that the program finishes sometime prior to the heat death of the universe. While this, unfortunately, makes this prototype infinitely worse than a true TeX-to-TikZ implementation, we view this as a necessary evil due to constraints of the average human lifespan (combined with the desire to, presumably, write more than zero papers in their life)[2].

Through intensive research, we found that this increases conversion time proportionally to the document's resolution[3][4] times the number of pages in the document. To help reduce this problem, we implemented an innovative optimization in this space: omitting pixels that are the same color as the background. We found that this helped to dramatically reduce compile times while still ensuring document quality and that researchers have to wait comically long before the document is ready for viewing.

[1] This is not to be confused with it being unfounded; I simply lacked the resources to put together a grant proposal in the first place.

[2] On second thought, that might not actually be such a bad thing as it would allow researchers unlimited free time. See `https://xkcd.com/303/`

[3] Its just some magic number that the PDF library I'm using seems to spit out. I'm sure using it is fine.

[4] On second thought, it might not be a magic number; I should probably read the documentation prior to publishing... Eh, no one will cite this anyways. After all, the thesis will be a much better version with much more detail. Could probably include the whole 1,337 pg TeX-to-TikZ specification in there after all...

4 Results

A sample of this entire document typeset using the prototype of TeX-to-TikZ can be found in Appendix A. In addition, the code for this project can be found at [5].

5 Conclusion

TeX-to-TikZ is a promising new way to typeset documents. While future work is needed to determine the potential limitations of this technology, given its clear benefit over traditional typesetting techniques, extensive work must first be put into the creation of a more efficient TeX-to-TikZ compiler to help the research community embrace this pivotal new technique for communicating, collaborating, and working. In addition, we hope that this work will inspire others to explore the possibility of revolutionizing other fields with TikZ. These potential future directions include, but are not limited to, Beamer-to-TikZ, SVG-to-TikZ, PNG-to-SVG, and MP4-to-TikZ. In addition, we hope that others will pursue integrating this technology into applications such as Paint, Photoshop, Illustrator, and more.

References

[1] DuckDuckGo and Google seem to agree on this; I, however, couldn't find any references to this on google scholar.

[2] *FIXME: Cite tikz website when on ground. I dont have the budget for airplane wifi.*

[3] Alex Friedman. "TeX-to-TikZ". In: *Sigbovik 2023*. Ed. by Editors please feel free to insert your name here. 2023, p. $i^{-4}\pi$.

[4] Alex Friedman. "TeX-to-TikZ". In: *Sigbovik 2023*. Citing document prior to peer review. 2023, p. 1.

[5] Alex Friedman. *TeX-to-TikZ Source Code*. `https://github.com/ahfriedman/tex-to-tikz`. 2023.

[6] *TikZ - Wikipedia*. `https://en.wikipedia.org/wiki/TikZ`. 2023.

A Sample Paper Typeset in TeX-to-TikZ

TeX-to-TikZ

Alex Friedman, Charlotte Clark,
Natalie McClain, Shaw-leon Chen,
Connie Chau, Jake Backer,
N'yoma Diamond, and Rose Bohrer

This interstate paper (with work occurring in 7 different US states) is highly reviewed being edited, in part, at 27,000ft (Flight Level 270).

Acknowledgements

I would like to thank my coauthors for generously contributing their names to this paper. Without them, this paper would not have the same SHA256 hash as it has today. I hope they will all be able to go on and have illustrious careers without this critical research overshadowing their future contributions.

1 Introduction

As it stands, there are many limitations to how academic papers are typeset. In the standard typesetting tool, LaTeX, users lack granular control over the exact appearance of their text

```
! TeX capacity exceeded, sorry [main memory size=5000000].
```