HGFRR: Hidden Geographical Fractal Random Ring

Anonymous Authors

Abstract

This is the abstract.

1 Introduction

This is the introduction.

More fascinating text. Features[1] galore, plethora of promises.

2 Background and Motivation

This is the background and motivation.

More fascinating text. Features[2] galore, plethora of promises.

3 HGFRR Design

This is the design.

More fascinating text. Features[3] galore, plethora of promises.

4 Proof and Analysis

This is the proof and analysis.

More fascinating text. Features[4] galore, plethora of promises.

5 Implementation Details

This is the implementation details.

More fascinating text. Features[5] galore, plethora of promises.

6 Evaluation

This is the evaluation.

More fascinating text. Features[6] galore, plethora of promises.

7 Related Work

This is the related work.

More fascinating text. Features[7] galore, plethora of promises.

8 Conclusion

This is the conclusion.

More fascinating text. Features[8] galore, plethora of promises.

9 Acknowledgments

A polite author always includes acknowledgments. Thank everyone, especially those who funded the work.

Figure 1: Wonderful Flowchart

10 Template

```
int wrap_fact(ClientData clientData,
              Tcl_Interp *interp,
              int argc, char *argv[]) {
    int result;
    int arg0;
    if (argc != 2) {
        interp->result = "wrong # args";
        return TCL_ERROR;
    }
    arg0 = atoi(argv[1]);
    result = fact(arg0);
    sprintf(interp->result,"%d",result);
    return TCL_OK;
}
```

Now we're going to cite somebody. Watch for the cite tag. Here it comes [1]. The tilde character (~) in the source means a non-breaking space. This way, your reference will always be attached to the word that preceded it, instead of going to the next line.

11 This Section has SubSections

11.1 First SubSection

Here's a typical figure reference. The figure is centered at the top of the column. It's scaled. It's explicitly placed. You'll have to tweak the numbers to get what you want.

This text came after the figure, so we'll casually refer to Figure 1 as we go on our merry way.

11.2 New Subsection

It can get tricky typesetting Tcl and C code in LaTeX because they share a lot of mystical feelings about certain magic characters. You will have to do a lot of escaping to typeset curly braces and percent signs, for example, like this: "The `%module` directive sets the name of the initialization function. This is optional, but is recommended if building a Tcl 7.5 module. Everything inside the `%{, %}` block is copied directly into the output. allowing the inclusion of header files and additional C code."

Sometimes you want to really call attention to a piece of text. You can center it in the column like this:

`_1008e614_Vector_p`

and people will really notice it.

The noindent at the start of this paragraph makes it clear that it's a continuation of the preceding text, not a new para in its own right.

Now this is an ingenious way to get a forced space. `Real *` and `double *` are equivalent.

Now here is another way to call attention to a line of code, but instead of centering it, we noindent and bold it.

`size_t :  fread ptr size nobj stream`

And here we have made an indented para like a definition tag (dt) in HTML. You don't need a surrounding list macro pair.

> `fread` reads from `stream` into the array `ptr` at most `nobj` objects of size `size`. `fread` returns the number of objects read.

This concludes the definitions tag.

11.3 How to Build Your Paper

You have to run `latex` once to prepare your references for munging. Then run `bibtex` to build your bibliography metadata. Then run `latex` twice to ensure all references have been resolved. If your source file is called `usenixTemplate.tex` and your `bibtex` file is called `usenixTemplate.bib`, here's what you do:

```
latex usenixTemplate
bibtex usenixTemplate
latex usenixTemplate
latex usenixTemplate
```

11.4 Last SubSection

Well, it's getting boring isn't it. This is the last subsection before we wrap it up.

References

[1] STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, M. F., AND BALAKRISHNAN, H. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review 31*, 4 (2001), 149–160.

Notes

[1]Remember to use endnotes, not footnotes!
[2]Remember to use endnotes, not footnotes!
[3]Remember to use endnotes, not footnotes!
[4]Remember to use endnotes, not footnotes!
[5]Remember to use endnotes, not footnotes!
[6]Remember to use endnotes, not footnotes!
[7]Remember to use endnotes, not footnotes!
[8]Remember to use endnotes, not footnotes!