

Reproducibility review of: Experimental evaluation of using BLE beacon for outdoor positioning in GPS-denied environment

Daniel Nüst 

2022-06-10



This report is part of the reproducibility review at the AGILE conference. For more information see <https://reproducible-agile.github.io/>. This document is published on OSF at <https://osf.io/8b7mr/>. To cite the report use

Nüst, Daniel. 2022. "Reproducibility Review of: Experimental Evaluation of Using BLE Beacon for Outdoor Positioning in GPS-Denied Environment." OSF. <https://doi.org/10.17605/osf.io/8b7mr>

Reviewed paper

Qamaz, Y., Schwering, A., and Bistron, J.: Experimental evaluation of using BLE beacon for outdoor positioning in GPS-denied environment, AGILE GIScience Ser., 3, 13, <https://doi.org/10.5194/agile-giss-3-13-2022>

Summary

The article present an experimental evaluation of a positioning system. Naturally, the physical/practical experiments could not be reproduced as part of this check of the computational reproducibility. However, the one figure in the article that is based on the data from the experiment, as well as the software for running the experiment shown in screenshots in the paper could be reproduced by me. With respect to the computational aspects and data visualisation, the given paper is reproducible.

Reproducibility reviewer notes

After a brief initial evaluation and contact, the authors quickly provided updated links and documentation to the required software and data, and even updated the manuscript accordingly.

Figure 4

Data and code for reproducing Fig. 4 in Section 4.4 of the paper are available at <https://doi.org/10.17605/OSF.IO/E48HX>. The OSF project only includes the GitHub repository at <https://github.com/YouQam/experimental-evaluation-ble-beacon> which I forked into the Reproducible AGILE organisation at the start of the reproducibility review: <https://github.com/reproducible-agile/experimental-evaluation-ble-beacon>

From there, the following steps renders the given Rmd document successfully:

```
git clone https://github.com/reproducible-agile/experimental-evaluation-ble-beacon
```

```
rmarkdown::render("experimental-evaluation-ble-beacon/Qamaz-et-al-2022.Rmd")
```

```
##
##
## processing file: Qamaz-et-al-2022.Rmd

## |                                | 0% |
## ordinary text without R code
## |
## label: global_options (with options) | 20%
## List of 1
## $ include: logi FALSE
## |
## ordinary text without R code
## |
## label: unnamed-chunk-3 | 40%
## |
## ordinary text without R code
## |
## label: unnamed-chunk-4 | 60%
## |
## ordinary text without R code
## |
## label: unnamed-chunk-5 | 80%
## |
## ordinary text without R code
## |
## label: unnamed-chunk-6 (with options) | 100%
## List of 1
## $ out.width: chr "100%"

## output file: Qamaz-et-al-2022.knit.md

## /usr/lib/rstudio-server/bin/quarto/bin/pandoc -RTS -K512m -RTS Qamaz-et-al-2022.knit.md --to gfm-yaml_metadata_block --from markdown+autolink_bare_uris+tex_math_single_backslash --output Qamaz-et-al-2022.knit.md
## /usr/lib/rstudio-server/bin/quarto/bin/pandoc -RTS -K512m -RTS Qamaz-et-al-2022.knit.md --to html4 --from gfm --output Qamaz-et-al-2022.html --standalone --self-contained --highlight-style pygments --te

##
## Preview created: /tmp/RtmpcsMnEH/preview-1c545c5a174.html

##
## Output created: Qamaz-et-al-2022.md

sessionInfo()

## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8 LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8 LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8 LC_NAME=C
##  [9] LC_ADDRESS=C LC_TELEPHONE=C
```

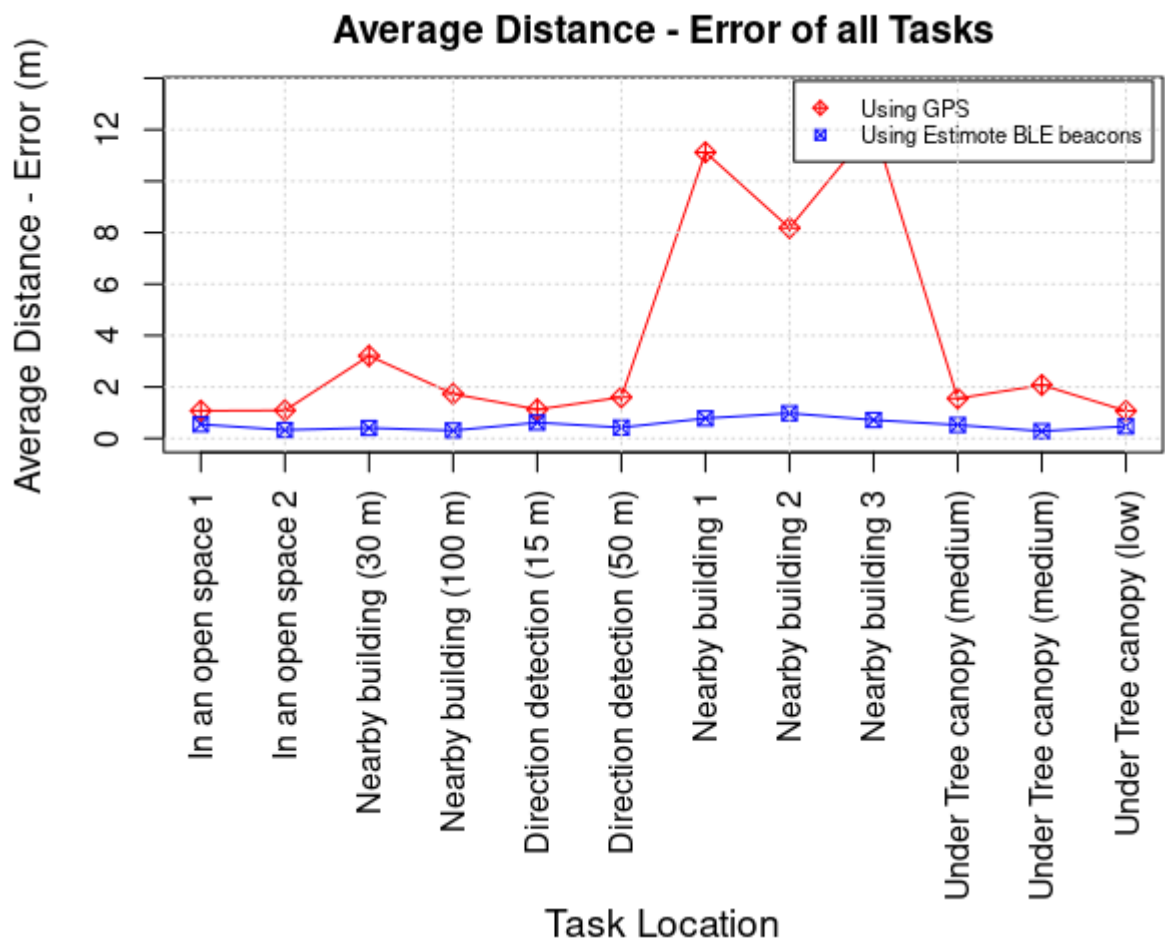
```
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] codetools_0.2-18 tinytex_0.37    markdown_1.1    rmarkdown_2.13
##
## loaded via a namespace (and not attached):
## [1] digest_0.6.29  magrittr_2.0.2 evaluate_0.15   rlang_1.0.2
## [5] stringi_1.7.6  cli_3.2.0      rstudioapi_0.13 tools_4.1.3
## [9] stringr_1.4.0  xfun_0.30      yaml_2.3.5     fastmap_1.1.0
## [13] compiler_4.1.3 htmltools_0.5.2 knitr_1.37
```

The created document includes a figure that matches Figure 4 with the exception of plot margins and therefore the legend overlaps the graph. It is shown below.

```
file.copy("experimental-evaluation-ble-beacon/Qamaz-et-al-2022_files/figure-gfm/unnamed-chunk-4-1.png",
"figure4.png", overwrite = TRUE)
```

```
## [1] TRUE
```

```
knitr::include_graphics("figure4.png")
```



The provided Binder link does work, but only after manually installing missing packages, because the given `install.r` file seems not to be picked up (cf. [pull request](#)). Then the created output is the same as in my own environment.

Experiment game

The given paper collects data with human participants based on a game call GeoGami, and the used version is available in a branch on GitHub at https://github.com/origami-team/origami/tree/ibeacon_integration_v3. Compared to the regular codebase (https://github.com/origami-team/origami/compare/ibeacon_integration_v3) the branch does indeed seem to include code for communicating with beacons and creating a game variant based on beacons. *Without access to the beacon hardware, it's not possible for me to fully evaluate this part of the research.* However, I'll see if I can build and run the game software at least.

The README includes documentation for building the app locally and for mobile operating systems. I try the former, though the documentation is very sparse and stops with the install, lacking information on build and usage.

```
git clone https://github.com/origami-team/origami
cd origami
git checkout ibeacon_integration_v3
```

```
$ docker run --rm -it -v $(pwd):/origami node:16-buster /bin/bash

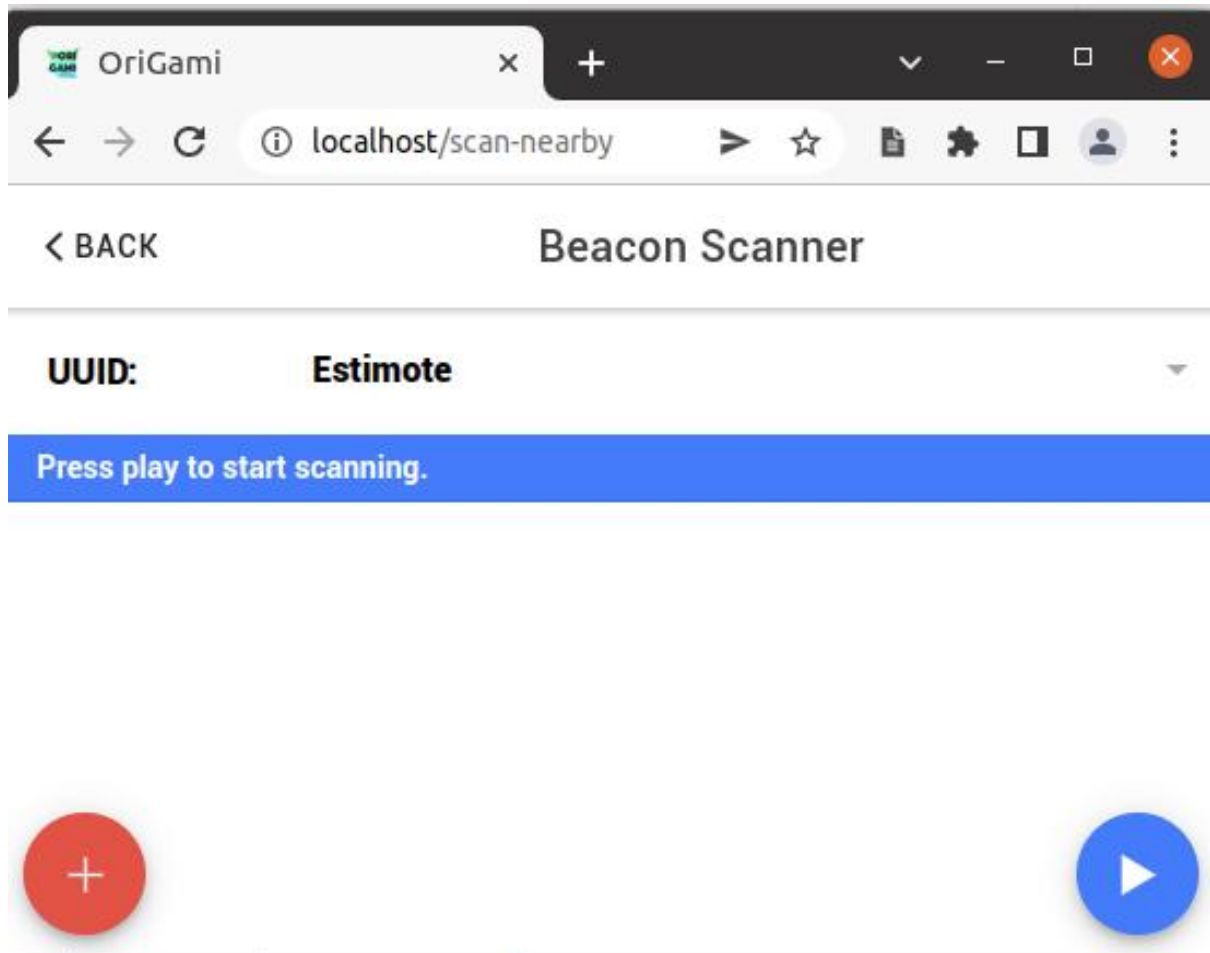
# in the container (outputs abbreviated):
cd /origami/
yarn install
# [...]
yarn run build-browser
# [...]
```

There were problems with using the latest Node.js, 18, but switching back to 16 did the trick: The `build-browser` script in the `package.json` created a directory `www`. Exploring this with a webserver:

```
docker run --rm -p 80:80 -v $(pwd)/www:/usr/share/nginx/html nginx
```

Gives me the UI of the GeoGami game, including some settings for Beacons.

```
knitr::include_graphics("geogami-screenshot-scan.jpg")
```



Without access to the required hardware, there is no point in evaluating the app further, or to test builds for mobile operating systems. The code and the UI though show that a system as described in the paper seems to have been actually built.

Comments to the authors

- Good job, using Binder!
- Please consider creating a registration for your OSF project - only then the files from the GitHub repo are properly archived, see <https://osf.io/e48hx/registrations>
- Add GeoGami to the OSF project (before creating a registration) so that tool is also archived
- The “MIT” license given in the GitHub repository is not suitable for data; see, e.g., <https://opendatacommons.org/faq/licenses/>
- Fix your plot margins and figure size in the code
- GeoGami should have a proper citation (i.e., archive to Zenodo/Software Heritage and add suggested citation in README)
- Take advantage of the fact that you have an OSF repo and add additional information there, e.g., a screencast/video of the app or more screenshots than you could fit into the paper.