Artifact Evaluation for Paper #363

1 INTRODUCTION

The artifacts evaluation process aims to validate the reproducibility and applicability of the findings outlined in our paper, "Plug-and-play Indoor GPS Positioning System with the Assistance of Optically Transparent Metasurfaces". The evaluation includes the following aspects:

Getting Started Guide. This section simply describes the system environment settings and the necessary environment dependencies demanded for the artifact.

Step-by-Step Instruction. This section describes the details on how to execute the submitted artifact and demonstrate the expected outcomes of the artifact.

1.1 Delivery

The necessary project files are attached along with this document. We also make the project available publicly in our Git repository. It could be found at the below link:

1.2 The fundamental concept of the submitted artifact

The artifact focuses on the metasurface coding of GPS signals in a real-time manner. Aiming to provide indoor GPS signal coverage, the system tracks the GPS satellites by predicting the Angle of Arrival(AoA) of the incident GPS signal, and lets the beam sweeps across the indoor area for continuous coverage. We first provide a bash script to fetch the almanac of GPS satellites provide publicly. Then the submitted code is able to interpret the file and predict the satellite locations and AoA accurately. Finally, the code provides a coding scheme in real time which lets the beam sweep across the coverage area.

2 GETTING STARTED GUIDE

2.1 Environment Setup

The submitted code for execution necessitates the following initial setup to ensure results as closely aligned as feasible with those depicted in the paper.

- Hardware i7-13700F@2.1GHz CPU or higher
- Operating System: Win10 or higher
- Python 3.8 or higher
- Basic python package for analysis including NumPy, Pandas

2.2 Almanac Download

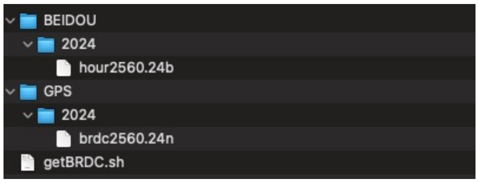
The system only requires a single almanac to operate. To simplify the downloading procedure, we provide a bash script to download the almanac files and put them in the corresponding locations. You could simply enter the path where the bash script "getBRDC.sh" is by enter the following command in a terminal:

```
cd YOUR_PREFIX_PATH/BRDC
```

Then you could execute the download script by the following command:

```
sh getBRDC.sh
```

The script will download the latest almanac for GPS satellites and BEIDOU satellites, unzip them and place them in the corresponding folder by default. Note that the script downloads the almanac file of the previous day and the system predicts the satellite locations in a real-time manner by default. If the downloads fail, you could still operate the coding procedure leveraging the almanac we provide for prediction, which will be elaborated in the following Step-by-step Instruction.



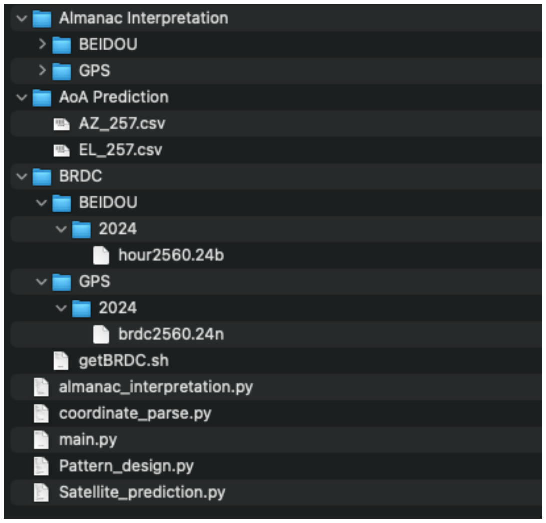
```
BEIDOU
  2024
    hour2560.24b
GPS
  2024
    brdc2560.24n
getBRDC.sh
```

3 STEP-BY-STEP INSTRUCTION

To demonstrate how we predict the satellite locations to generate the phase configuration of metasurfaces in the paper, we provided a code example in the attachment and also on Github. Each parameter in the code will be introduced along with its meaning and usage in the following instructions.

3.1 Extracting the files

After extracting the code files, you will find the files as follows: You could find the "main.py" which is the "main" file for program execution. Open this file in your own editor.



```
Almanac Interpretation
  BEIDOU
  GPS
AoA Prediction
  AZ_257.csv
  EL_257.csv
BRDC
  BEIDOU
    2024
      hour2560.24b
  GPS
    2024
      brdc2560.24n
  getBRDC.sh
almanac_interpretation.py
coordinate_parse.py
main.py
Pattern_design.py
Satellite_prediction.py
```

3.2 Fundamental Parameter Settings.

In our code files, several parameters are provided for specific customization. Each parameter is introduced individually as follows:

- location: describes the location of the metasurfaces deployment demanding no precision. A BLH coordinate of the building, for example, could be fine (default [0,0,0]).
- window_orientation: describes the window orientation, which is the angle in degree between the orientation and the north direction in clockwise (default 90 degrees, which indicates the window faces east).
- offline_flag: sets True if the download fails for an offline demonstration (default False for demonstration in real time).

3.3 Results of the Execution

After setting all parameters, running "main.py" will generate the metasurface phase pattern of the metasurfaces in real time. An figure updates in time will be generated to demonstrate the phase pattern of the metasurfaces. Fig. 1 demonstrate an example. We pause the plotted figures for 0.1s every update for better demonstration. Note that you need to terminal the python process to stop the system

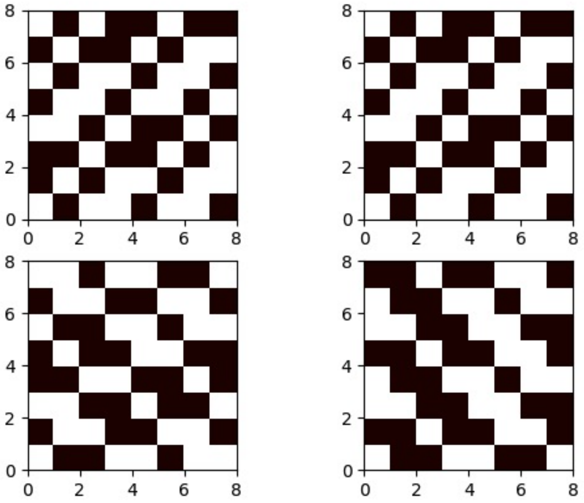


Figure 1: Phase Pattern Demonstration

4 CONCLUSION

We would like to express my sincere appreciation for dedicating your time and effort to review the artifacts of "Plug-and-play Indoor GPS Positioning System with the Assistance of Optically Transparent Metasurfaces." For further questions and explanations, please feel free to contact us without hesitation.