Submission ID #: 61540
Scriptwriter Name: Bridget Colvin
Project Page Link: https://www.jove.com/account/file-uploader?src=18769748

Title: Simulating Imaging of Large-Scale Radio Arrays on the Lunar Surface

Authors and Affiliations: Alexander M. Hegedus

Department of Climate and Space Sciences and Engineering, University of Michigan

Corresponding Author:
Alexander M. Hegedus
alexhege@umich.edu

Author Questionnaire

1. Microscopy: Does your protocol demonstrate the use of a dissecting or stereomicroscope for performing a complex dissection, microinjection technique, or similar? **N**

2. Software: Does the part of your protocol being filmed demonstrate software usage? **Y**

3. Filming location: Will the filming need to take place in multiple locations (greater than walking distance)? **N**

Protocol Length
Number of Shots: **24**

Introduction

1. Introductory Interview Statements

REQUIRED:

1.1. **Alex Hegedus**: This package extends the industry standard simulation software for Radio Astronomy, CASA, for the simulation of Lunar based arrays, a re-emerging field of interest with many unique scientific applications **[1]**.

1.1.1. INTERVIEW: Named talent says the statement above in an interview-style shot, looking slightly off-camera

REQUIRED:

1.2. **Alex Hegedus**: Our protocol can utilize a combination of astronomical charts from NASA's SPICE package alongside elevation maps of the Moon's surface from Lunar Reconnaissance Orbiter to accurately simulate any array on the Moon **[1]**.

1.2.1. INTERVIEW: Named talent says the statement above in an interview-style shot, looking slightly off-camera

Protocol

2. Array Configuration Creation

2.1. Before beginning a simulation, navigate to the Deep Blue Data website **[1-TXT]** and download the software package **[2]**.

2.1.1. WIDE: Talent opening page **TEXT:** **https://deepblue.lib.umich.edu/data/concern/data_sets/bg257f178?locale=en**

2.1.2. SCREEN: screenshot_1 *Video Editor: please speed up*

2.2. The software has only been tested in a UNIX environment and may not fully function in other environments **[1]**.

2.2.1. SCREEN: screenshot_2: 00:00-00:05

2.3. To customize the createArrayConfig.py script, provide a list of longitude and latitude coordinates for each antenna to select the configuration of the array and change the LunarPath variable in the script to reflect the new download location of the Digital Elevation Model containing the elevation data of the lunar surface **[1]**.

2.3.1. SCREEN: screenshot_3 *Video Editor: please speed up*

2.4. Use the command to run the createArrayConfig.py script to use the lunar Digital Elevation Model to solve for the elevation at each longitude and latitude for each antenna **[1]**.

2.4.1. SCREEN: screenshot_4

2.5. Save the longitude, latitude, and elevation to files and print to the screen for easy copying and pasting into the next script **[1]**.

2.5.1. SCREEN: screenshot_5

2.6. Then make figures showing the array configuration on top of the local lunar topography **[1]**.

2.6.1. SCREEN: screenshot_6

3. **Coordinate Alignment Using SPICE**

3.1. To customize the eqArrOverTimeEarth.c script, copy the longitude, latitude, and each antenna elevation output into the corresponding lists in the script **[1]** and update the **num-spacecraft** variable with the number of receivers and corresponding coordinates **[2]**.

3.1.1. WIDE: Talent copying output, with monitor visible in frame
3.1.2. SCREEN: screenshot_7 *Video Editor: please speed up*

3.2. Update the lunar_furnsh.txt included in the package with the new path names for the required frame and ephemeris files and specify the set of dates on which the observations should occur to inform the ephemerides within SPICE to accurately track where the Earth and Sun are in relation to the defined array for those dates **[1-TXT]**.

3.2.1. SCREEN: screenshot_8 *Video Editor: please speed up* **TEXT: 48 dates occurring roughly weekly throughout 2025 present in script**

3.3. Specify the targeted area of the sky for the array to track and image **[1-TXT]**.

3.3.1. SCREEN: screenshot_9 **TEXT: Script preset to save RA Dec of Earth as seen from lunar surface**

3.4. Next, use the **gcc** command to compile the eqArrOverTime.c script and change the paths to reflect where the cspice libraries are located **[1-TXT]**.

3.4.1. SCREEN: screenshot_10 *Video Editor: please speed up* **TEXT: Caution: link correct SPICE headers**

3.5. Use the command to run the equatorial array over time executable to obtain a number of files, each with a set of variables in them. Most important are the XYZ position of each antenna in the J2000 coordinates and the Right Ascension and Declination coordinates of the targeted area in the sky **[1]**.

3.5.1. SCREEN: screenshot_11 *Video Editor: please emphasize XYZ position in J2000 coordinates and RA and Dec coordinates when mentioned*

3.6. Then save the output variables to .txt files containing the data for all the requested dates **[1]**.

3.6.1. SCREEN: screenshot_12

4. Array Response Simulation with Common Astronomy Software Applications (CASA)

4.1. To customize the LunarEarthPicFreqIntegration.py script, specify the observing frequency for the array at which to make an image **[1]** and specify a CASA (cah-sah) compatible truth image with Jansky-pixel values for the array to reconstruct **[2]**.

4.1.1. WIDE: Talent specifying observing frequency, with monitor visible in frame
4.1.2. SCREEN: screenshot_13 *Video Editor: please speed up*

4.2. Change the constants in the code to reflect the size and resolution of the input truth image **[1]**.

4.2.1. SCREEN: screenshot_14 *Video Editor: please speed up*

4.3. Use the command to run the LunarEarthPic.py script. The **-numSC** flag is used to inform the code how many antenna and/or receivers are being used and helps unpack the data from the .txt files containing the receiver coordinates **[1]**.

4.3.1. SCREEN: screenshot_15 *Video Editor: please emphasize -numSC flag when mentioned*

5. Noiseless and Noisy Data Imaging

5.1. To customize the noiseCopies.py script, set the System Equivalent Flux Density **[1]** and set the bandwidth being integrated over in variable **noise** line 200 to 500 kilohertz **[2]**.

5.1.1. WIDE: Talent setting SEFD, with monitor visible in frame
5.1.2. SCREEN: screenshot_16

5.2. Set the integration time in variable **noise** line 200 and use the command to run the noiseCopies.py script. The script will first create an image from the noiseless visibility data, calling standard radio astronomy algorithm CLEAN (clean) to create the image **[1]**.

5.2.1. SCREEN: screenshot_17 *Video Editor: can speed up*

5.3. The script will then create copies of the measurement set and add the appropriate noise level to the complex visibility data before using CLEAN to image the data for a range of integration times up to 24 hours and over several robust weighting scheme values **[1]**.

5.3.1. SCREEN: screenshot_18

5.4. Depending on the configuration of the array, the image quality may vary with the choice of data weighting schemes **[1]**.

5.4.1. SCREEN: screenshot_19

Protocol Script Questions

A. Which steps from the protocol are the most important for viewers to see?

n/a

B. What is the single most difficult aspect of this procedure and what do you do to ensure success?

Making sure all the libraries are in order is crucial; not linking the correct SPICE headers in step 3.4 is an easy mistake to make. Also changing the constants in the code to reflect the size and resolution of the input truth image in step 4.2 may be easy to mess up if one does not follow the instructions closely.

Results

6. Results: Representative Large-Scale Radio Array Lunar Surface Imaging Simulation

6.1. Running create array config.py as demonstrated should create an elevation map similar to that presented **[1]**, in which the configuration of the defined array **[2]** is plotted on top of the local topography of the lunar surface as derived from the Lunar Reconnaissance Orbiter Lunar Orbiter Laser Altimeter-derived Digital Elevation Model **[3]**.

6.1.1. LAB MEDIA: Figure 1
6.1.2. LAB MEDIA: Figure 1 *Video Editor: please emphasize array (stars and central black circle)*
6.1.3. LAB MEDIA: Figure 1 *Video Editor: please emphasize topography under array*

6.2. Using CASA to simulate an array response should result in a similar output to that observed here **[1]**, which can be used to calculate the visibility data **[2]**.

6.2.1. LAB MEDIA: Figure 2 *Video Editor: please data in center of graph*
6.2.2. LAB MEDIA: Figure 2

6.3. Data imaging **[1]** can then generate noiseless **[2]** and noisy images **[3]**, with the noisy images appearing less clear **[4]** than the noiseless images **[5]**.

6.3.1. LAB MEDIA: Figures 3 and 4
6.3.2. LAB MEDIA: Figures 3 and 4 *Video Editor: please emphasize Figure 4*
6.3.3. LAB MEDIA: Figures 3 and 4 *Video Editor: please emphasize Figure 3*
6.3.4. LAB MEDIA: Figures 3 and 4 *Video Editor: please emphasize light blue/green/yellow background of Figure 4*
6.3.5. LAB MEDIA: Figures 3 and 4 *Video Editor: please emphasize central data of Figure 3*

Conclusion

7. Conclusion Interview Statements

7.1. **Alex Hegedus**: This protocol is a great tool for prototyping arrays for specific scientific targets. Small arrays work great for bright transients, like solar radio bursts, and larger arrays work for dimmer planetary emissions **[1]**.

7.1.1. INTERVIEW: Named talent says the statement above in an interview-style shot, looking slightly off-camera