Installing and Running the Space Telescope Image Product Simulator to Generate Mock WFIRST Images

Benjamin F. Williams and Rubab Khan (U. Washington)
Brian York (STScI)

This is a brief cookbook for installing, setting up, and using the Space Telescope Image Product Simulator (STIPS) to produce simulated WFIRST images. Setting up STIPS for local use takes some time and disk space for all of the downloads, but this document should make it a straight-forward process. All of these setup commands should be done in a bash terminal window. All commands are in TrueType font.

Step 1: Install Conda for Python 3 Using a Bash Shell

```
bash
```

Go to https://conda.io/miniconda.html
Download the appropriate installer for Python 3.

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

Run the installer to install Python 3 with and Conda

```
chmod +x ./Miniconda3-latest-Linux-x86_64.sh
./Miniconda3-latest-Linux-x86_64.sh
```

Accept terms and choose install location

Add the installation to your PATH

```
export PATH=[path to miniconda3]/bin:$PATH
```

Step 2: Add STScI to your Conda configuration

```
conda config -add channels http://ssb.stsci.edu/astroconda
```

Step 3: Create the STIPS environment using Conda

```
conda create -n STIPS python=3.6.6 astropy=1.3.2 numpy=1.12.1 scipy=0.18.1
photutils=0.3.2 pysynphot=0.9.8.5 webbpsf=0.6 #This can take many minutes to run
```

Enter yes to proceed

Step 4: Activate the environment

```
source activate STIPS
echo $WEBBPSF_PATH # This should return a path if all has gone well to this point
```

Step 5: Obtain the necessary cdbs and pandeia data files

```
wget ftp://archive.stsci.edu/pub/hst/pysynphot/synphot1.tar.gz
tar xvfz synphot1.tar.gz
cd grp/hst
wget -recursive ftp://ftp.stsci.edu/cdbs/grid #large download
wget -recursive ftp://ftp.stsci.edu/cdbs/calspec
wget -recursive ftp://ftp.stsci.edu/cdbs/extinction
wget http://ssb.stsci.edu/pandeia/engine/1.0/pandeia_data-1.0.tar.gz
tar xvfz pandeia_data-1.0.tar.gz
```

Step 6: Put everything in the right place and set environment paths for cdbs and pandeia

```
mv ftp.stsci.edu/cdbs/grid [path to synphot]/grp/hst/cdbs/
mv ftp.stsci.edu/cdbs/calspec [path to synphot]/grp/hst/cdbs/
mv ftp.stsci.edu/cdbs/extinction [path to synphot]/grp/hst/cdbs/
rm -r ftp.stsci.edu
export PYSYN_CDBS="[path to synphot]/grp/hst/cdbs"
export pandeia_refdata="[path to pandeia data]/pandeia_data-1.0"
```

Step 7: Pip install dependencies

```
pip install "esutil==0.6.0"
pip install "montage-wrapper==0.9.9"
pip install "jwst_backgrounds==1.1.1"
pip install "pandeia.engine==1.0"
pip install "poppy==0.6.1"
```

Step 8: Clone the STIPS GitHub repositories

```
cd [path where you want STIPS installed]
git clone https://github.com/spacetelescope/STScI-STIPS-UI.git
git clone https://github.com/spacetelescope/STScI-STIPS.git
```

Step 9: Set STIPS data path in environment

```
export stips_data="[path to STScI-STIPS-UI]/STScI-STIPS-UI/sim_input/stips_data"
```

Step 10: Install STIPS

```
cd STScI-STIPS
python setup.py install
```

Assuming the installation then proceeds without error, now try the example by following the next few steps.

Step 11: Make a new directory to work in (outside of source code tree). In this example, we make a directory off of the home directory

```
mkdir ~/testSTIPS
```

Step 12: Copy the test input source list and python script

```
cp [path to STScI-STIPS]/STScI-STIPS/examples/example_H158.cat ~/testSTIPS/
cp [path to STScI-STIPS]/STScI-STIPS/examples/example.py ~/testSTIPS/
```

Step 13: Run the example script

```
cd ~/testSTIPS/
python example.py
```

This will generate some temporary files while it is running. When it completes there will be a source list file, a PSF file, and an image file. Assuming this image looks acceptable (open it in ds9 and have a look), you can now generate any input list in the same format as the example, edit the example python script, and make your own custom WFIRST images. There are other examples one can try at:

https://github.com/spacetelescope/STScI-STIPS#stips-examples

Staying up to date

You can keep your STIPS code up to date using `git pull` in your STScI-STIPS directory. However, keeping dependencies up to date is a bit more complicated. The first modules to update would be poppy and webbpsf to version 0.7. To do this, you need to pip install the versions and also put the supporting data in the right place.

```
pip install "webbpsf==0.7.0"

cd $WEBBPSF_PATH

cd ../

wget http://www.stsci.edu/~mperrin/software/webbpsf/webbpsf-data-0.7.0.tar.gz
```

```
tar xvfz webbpsf-data-0.7.0.tar.gz
```

```
pip install "poppy==0.7.0"
```

Working Offline

If you need to run STIPS on a machine that does not have an active internet connection, you must install the backgrounds library on the machine locally. To do this, you need to download quite a bit of data and put it in the correct place. Here are the instructions: Download Sky Background Database Files:

```
wget http://archive.stsci.edu/missions/jwst/simulations/straylight/sl_cache/sl_cache_tar1_0000-0249.tgz
wget http://archive.stsci.edu/missions/jwst/simulations/straylight/sl_cache/sl_cache_tar2_0250-0499.tgz
wget http://archive.stsci.edu/missions/jwst/simulations/straylight/sl_cache/sl_cache_tar3_0500-0749.tgz
wget http://archive.stsci.edu/missions/jwst/simulations/straylight/sl_cache/sl_cache_tar4_0750-0999.tgz
wget http://archive.stsci.edu/missions/jwst/simulations/straylight/sl_cache/sl_cache_tar5_1000-1249.tgz
wget http://archive.stsci.edu/missions/jwst/simulations/straylight/sl_cache/sl_cache_tar6_1250-1499.tgz
wget http://archive.stsci.edu/missions/jwst/simulations/straylight/sl_cache/sl_cache_tar7_1500-1749.tgz
wget http://archive.stsci.edu/missions/jwst/simulations/straylight/sl_cache/sl_cache_tar8_1750-1966.tgz
```

Put all of the numbered files (0000, 0001, .., 1966) into a directory named 'remote_cache'

```
ls sl*tgz | awk '{print "tar xvfz",$1}'| sh
mkdir remote_cache
mv ????  remote_cache/
```

Put a file named VERSION into remote_cache. This file is not included in the downloads for some reason, but can be downloaded:

```
wget https://archive.stsci.edu/missions/jwst/simulations/straylight/sl_cache/VERSION
mv VERSION remote_cache/
```

Put 'remote_cache' into a directory named 'background'.

```
mkdir background
mv remote_cache background/
```

Put 'background' into your stips_data directory at the root level

```
mv background [PATH TO STScI-STIPS-UI]/sim_input/stips_data
```

In stips_data/background, put a directory named 'jbt_refdata'.

```
cd [PATH TO STScI-STIPS-UI]/sim_input/stips_data/background
mkdir jbt_refdata
```

In that directory, put a copy of all of the contents of the 'refdata' directory from the jwst_backgrounds python package.

```
cd jbt_refdata
cp [PATH TO miniconda3]/envs/STIPS/lib/python3.6/site-packages/jwst_backgrounds/refdata/* .
```

With these files in place, STIPS will run as it did before, but it will not look up background data online.