

Optimum Interpolation SST for AVHRR (version 2) READ_ME

Jan 12, 2012

THE CODE PACKAGE

This code package contains the scripts and Lahey FORTRAN codes used to operationally produce the daily OISST using in situ and satellite data on a Linux machine. The package is provided for public disclosure but there is no guarantee that these will work on other platforms or using other versions of FORTRAN.

AVHRR stands for the Advanced Very High Resolution Radiometer while AMSR here refers to the Advanced Microwave Scanning Radiometer on board the EOS platform Aqua.

The contents of the code tar file can be extracted using the command:

tar -xvf tarname

where tarname = file to extract, e.g., oisst_daily_avhrr_v02r00_src_r[yyyymmdd].tar.

There are 3 directories in the tar:

1. **src**- contains FORTRAN code
2. **script**- contains scripts used to compile and/or run codes
3. **docs** - contains relevant source code documentation including this

DATA FLOWCHART

This document describes the processing that uses satellite data from AVHRR only, but there is a similar product that uses AVHRR and AMSR data. The flowchart (Fig. 1) shows the steps in OISST processing, and in this case, the AMSR boxes should be ignored. Fig. 1 shows three parallel processing chains for the different input data types. All inputs must be prepared before interpolation can be performed. The three steps are:

- 1) ftp of input data (ice concentration, satellite SST, in situ SST) from remote sites,
- 2) quality checks and gridding of each input dataset on a $\frac{1}{4}$ degree grid separately,
- 3) bias correction (for in situ and satellite SSTs) and conversion to SST (for ice).

In reality, the codes are run serially, and there are some dependencies. That is, preparation of the ship and buoy SSTs needs to precede the bias adjustment of the satellite data. Once all the inputs datasets

have been bias-adjusted, they are passed to the optimum interpolation code. The binary output is then converted to the netCDF format, and transferred for distribution and archive.

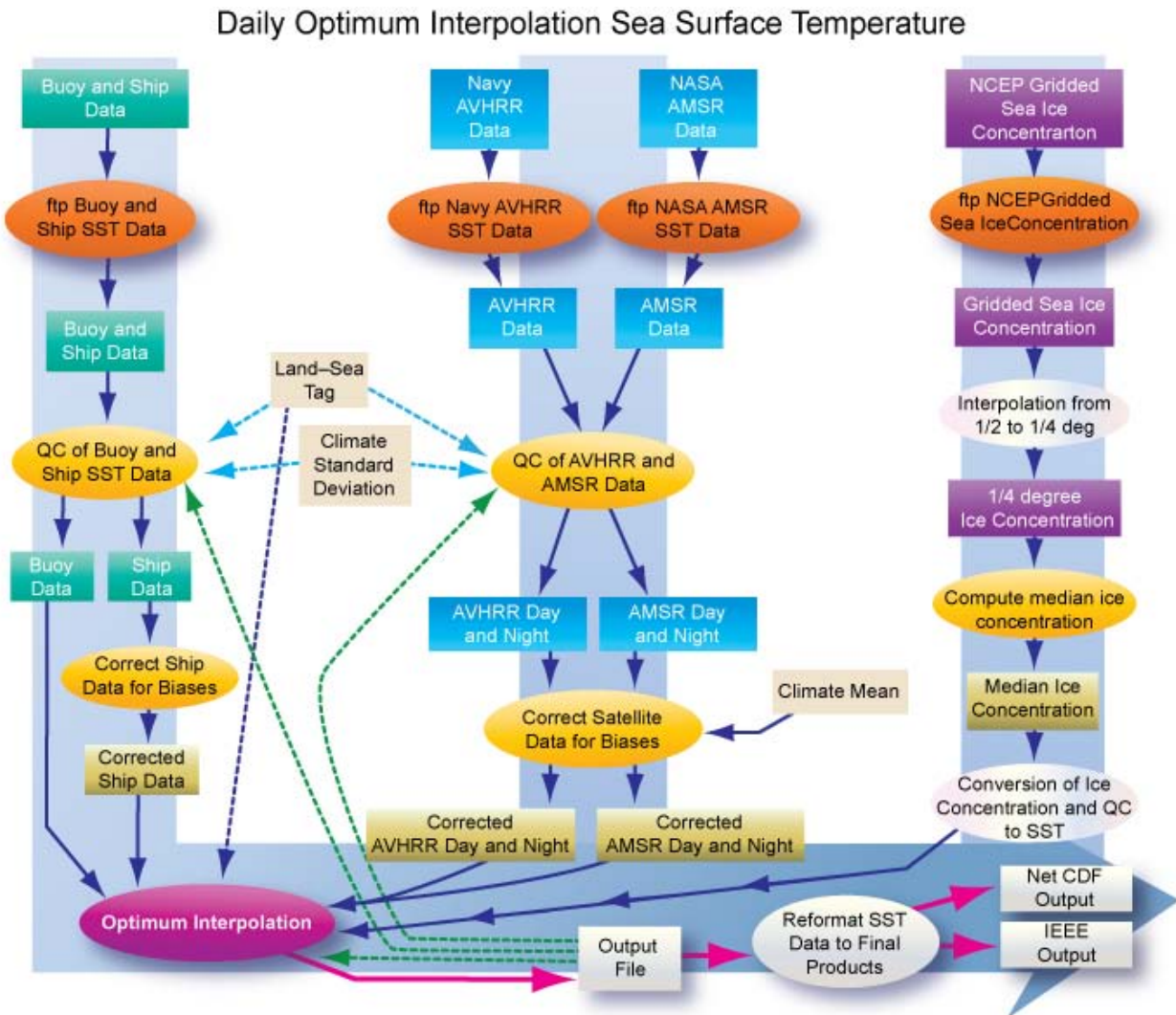


Figure1. OISST V 2.0 flowchart

CODES AND SCRIPTS

General. The FORTRAN codes used in the AVHRR-only and AVHRR+AMSR are essentially the same. The difference is in the main script, which determines which datasets to prepare and use for OI. One single script calls all programs and additional scripts needed for end-to-end processing for a single day. Fig 2 illustrates the relation between processing steps, codes, and data filenames.

There are two main processing OISST scripts for AVHRR-only:

- 1) **oisst_avhrr_only_data_intv2.sh** generates the “interim” or preliminary OISST product with a 1-day latency, and uses all the data available at run time.
- 2) **navy1718-finv2-oisst4-oper.sh** generates the “final” OISST with 15-day latency so that additional input data is included.

The reason for the two scripts is that there are users that need the product immediately. However, some computations involve a 15-day window centered on the day being processed. This window is truncated in the “interim” run because no data is available yet for the future 7 days.

The two main scripts use mostly the same FORTRAN codes and scripts, but the ftp and some other initial steps are skipped in the “final” run, since they have already been completed in the “interim” run. The “interim” processing is represented in the data flowchart (Fig. 2) since it goes through all the steps. The paths and filenames shown are for the “interim” script.

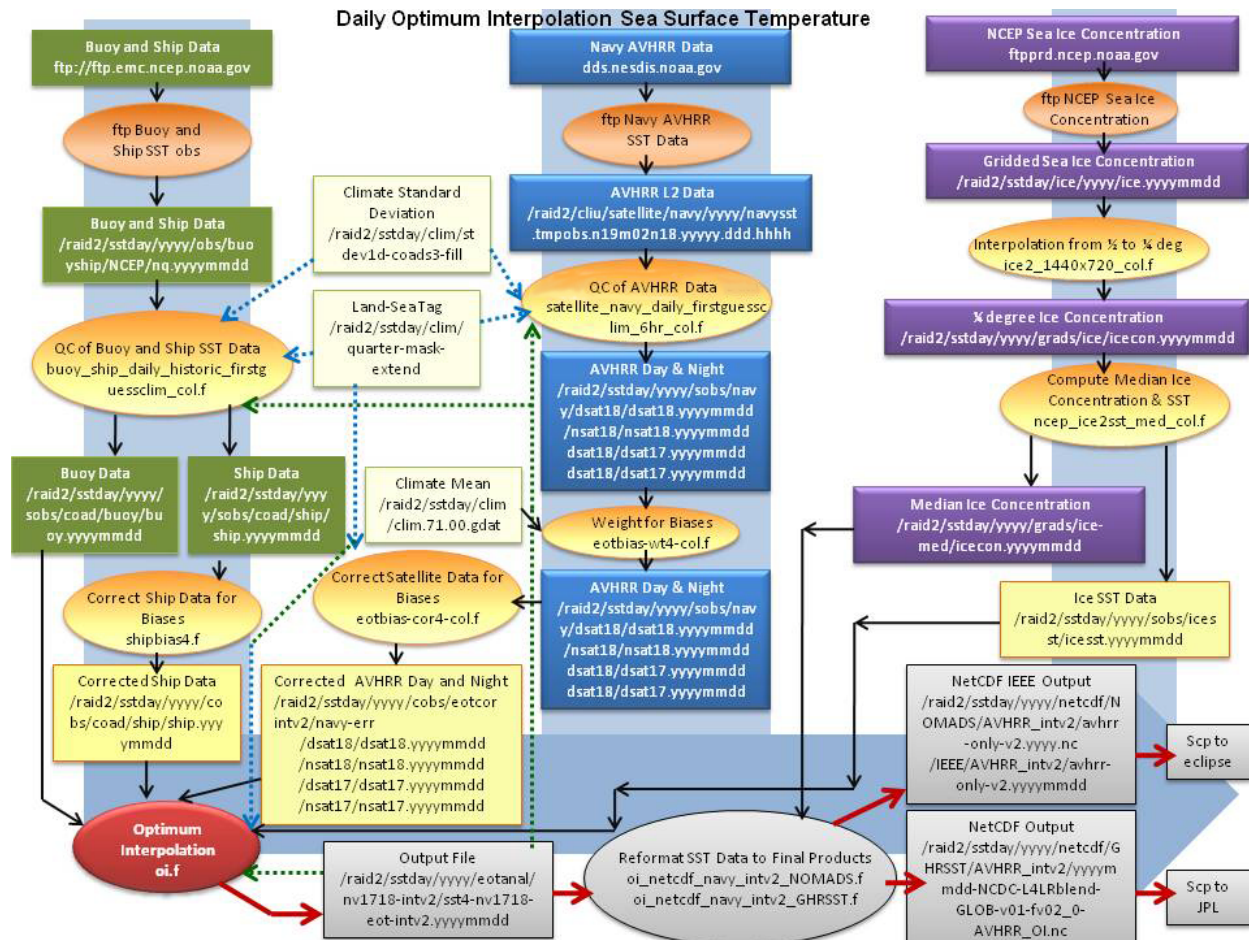


Figure 2. OISST V.2.0 data flowchart. Ovals show processing steps and associated FORTRAN code (ending with “.f”). Boxes show directory paths and filenames, where yyyy=year, mm=month and d=day of month.

Date to process. To determine the date to process, the main script reads a pre-existing text file containing the date last processed successfully, and increments the date by one. The path and filename containing the date is hardcoded in the main script, and must be changed to run on another machine:

oisst_avhrr_only_data_intv2.sh looks for the date in **/raid2/cliu/sstday/script/oi_intv2.txt**

navy1718-finv2-oisst4-oper.sh looks for the date in **/raid2/cliu/sstday/script/oi_finv2.txt**

When the script has successfully completed, the text file content is changed to the date just run. Thus, if any day needs to be rerun, then the date in the text file has to be manually set to the day prior. Based on this date, the script will also form the year, month and day strings.

Processing parameters. The main scripts wpass information required by the FORTRAN code by writing a text file ending with “.parm. Other than the date, the information in the parameter file can include directory paths, filenames, position in the filename string where the program may need to make changes. Computational values such as averaging weights, window size, rejection criteria are also passed using the parameter file. The format and order that the information is written out is critical.

BUILD/RUN INSTRUCTIONS

The two scripts above can be run by just typing: **./scriptname**. The assumption is that the paths have been correctly defined and that the FORTRAN executables exist. Note that the other scripts are called by the main script. The script named **compile_oisst_avhrr_only_intv2.sh** compiles and links all the FORTRAN code and libraries to make executables on a Linux machine. The FORTRAN compile command for each code may also be present in some scripts as comments. Most of the data transfer is done by the main script. Note that Lahey, not GNU Fortran is used. The code is a mix of heritage Fortran 77 and later.

The codes can also be compiled and run individually by issuing the individual compile commands in the compile script.