TerraFusion – Advanced Fusion project progress summary

Author: Yat Long Lo (Richie)

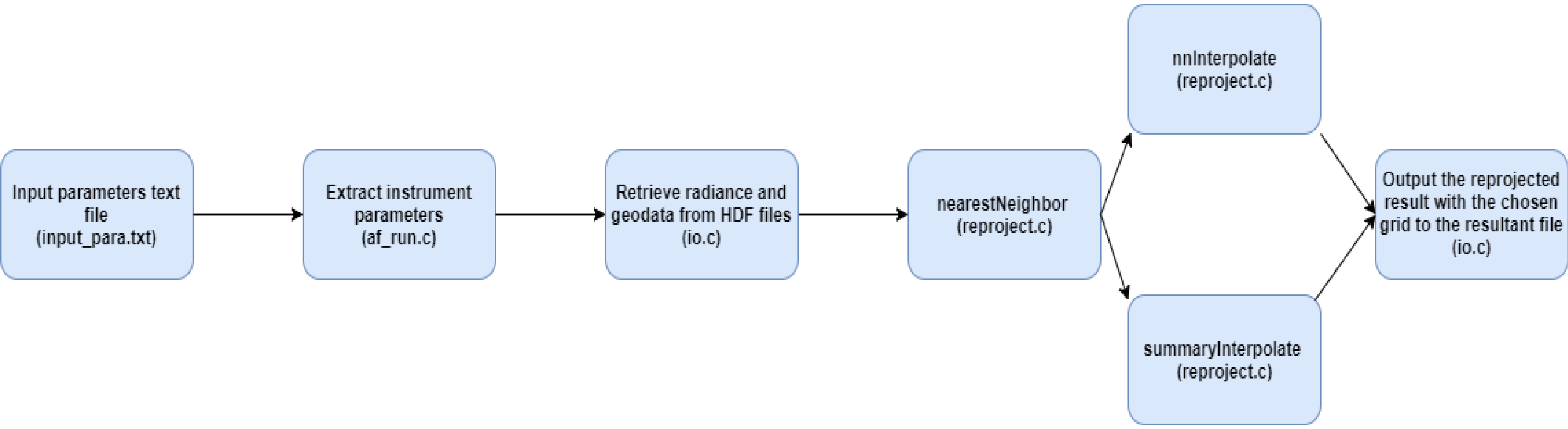
Email: yllo2@illinois.edu

Repository: https://github.com/TerraFusion/advancedFusion

1. Purpose

This document provides a progress summary for the advanced fusion portion of the TerraFusion project in the summer 2017 period of development. It will describe the current capabilities of the program, and look at what should be the next steps based on the groundwork that we have laid this summer. I hope this will be a comprehensive summary for the next developer to continue with the development.

2. General Flow


Input parameters text file (input_para.txt)
Extract instrument parameters (af_run.c)
Retrieve radiance and geodata from HDF files (io.c)
nearestNeighbor (reproject.c)
nnInterpolate (reproject.c)
summaryInterpolate (reproject.c)
Output the reprojected result with the chosen grid to the resultant file (io.c)

The figure about describes the general flow of advanced fusion program, at this stage of development. After careful consideration of different input methods, we have decided to go with an input text file for users to select their parameters. The text file requires certain format, in which an order of parameters has to be followed. All the parameters would be extracted by the af_run.c script. Based on the parameters, different functions would be called to retrieve data and perform reprojection. All the IO operations are done by the io.c script, while all the reprojection operations are done by the reproject.c script. Important details and features of each part will be given in the following sections.

3. Input method

An input text file method is chosen over the use of command line arguments due to the significant amount of arguments that a user has to specify to perform reprojection. The use of input text file would standardize the input format for the ease of processing.

For the program to run correctly now, the text file requires a particular order, starting with the path to the BasicFusion file and the output file. Then parameters of the instrument to project (project_instrument) should be next, followed by parameters of the instrument to be projected on (base_instrument).

4. Input Extraction and subsequent function calls

   af_run.c is responsible for input extraction and a cascade of function calls of I/O and reprojection operations. Necessary checks have been implemented to ensure the presence and validity of necessary arguments. Errors should be returned if parameters are missing or the order is not followed. The extraction process only supports ASTER, MODIS and MISR. Corresponding functions calling logic is only implemented for MISR on MODIS due to our limited logic on different use cases

   Next steps:

   1. Refactoring effort should be made on the code for the input extraction part.
   2. More cases should be added for different reprojection based on the specified input parameters identified from the extraction part, like ASTER on MODIS.

5. I/O operations

   io.c is responsible for all I/O operations one may need to perform reprojection. It supports all 5 instruments of the TERRA satellite, with variations among them. For instance, downsampling would be performed on MISR radiance data during retrieval if needed based on user specifications. In terms of granularity, functions are developed for instruments like MODIS support retrieval by bands, which are then stitched together granule after granule. Detailed description of each function can be found in the script itself.

   Next steps:

   1. There are no standard output functions for the reprojection results, only sample output functions have been written to test for MISR on MODIS, latitude data and longitude data. More than 1 output functions may be needed due to differences in formatting in different cases of reprojection.
   2. Special handling function for ASTER may be required as the granules cannot be stitched together in the raw form. Rotational properties have to be considered before stitching, given that the dimensions of each granule are not the same.
   3. To increase efficiency for I/O operations, most existing functions have been refrained from the use of realloc by calculating the size of the final dataset in the very beginning. However, not every function has been under such treatment, which should be a quick fix.

6. Reprojection operations

   All reprojection operations are in the reproject.c script, developed by YiZhao Gao. Description of each function exists in the header file of the script.

   Next steps:

   1. Reprojection operations account for most of the computation time. Due to the lack of dependencies between calculations, the performance would highly benefit from parallelization with tools like MPI
   2. More reprojection functions will be added in the future to accommodate different needs
   3. The commenting style and code descriptions of the script should coherent with the rest of TerraFusion project, like programs written in the BasicFusion part of the project.

7. Test cases

Several test cases have been developed by YiZhao and me in the process of development to test out different functions. For instance, a test script was developed to test the summaryInterpolate function by projecting one granule of ASTER data onto the MODIS grid. These scripts would be a nice starting point to see how the pieces come together and how the future development direction should be, by making use of the existing framework. These test cases can be built from the Makefile, like the core program.