Parallel Structure Factor

Yongchang JI (yji@cs.ucf.edu)

(Last update October 29, 2004)

CONTENTS

A. INTRODUCTION

Parallel Structure Factors (PSF) program is used to compare two 3D density maps and get some factors between these two data sets, so we can know the similarity of these two maps. PSF use as input: two 3D density maps generated by P3DR, output: some factors such as Standard Rfactor, Rfactor for Phase, Phase Difference, and Correlation Coefficient. The input 3D maps have been stored in the Purdue *.PIF format.

PSF is used in conjunction with our parallel 3D reconstruction algorithm (P3DR) in Cartesian coordinates for objects without symmetry.

Contact us: yji@cs.ucf.edu, dcm@cs.ucf.edu

B. PROGRAM INPUT

Input parameters of PSF are as follows :

---

map1filename	(C)
map2filename	(C)
res_hi, res_low, res_step, pixel_size	(F, F, I, F)

---

1. map1filename (C)

Specify the name of the PIF format 3D density map to be used for compared.

2. map2filename (C)

Specify the name of the PIF format 3D density map to be used for compared.

3. res_hi, res_low, res_step, pixel_size (F, F, I, F)

pixel_size is the pixel size for these two maps, if they are not the same as the pixel size in the map file, it will give a warning, and using the input pixel size.

res_hi and **res_low** specify the resolution range to which we compare these two maps, and the regions beyond this range will not be compared. **res_hi** is the upper radius resolution of this range, and it can't be set to a value any SMALLER than twice the size of the IMAGE pixels (the so-called Nyquist limit imposed because the Fourier transform of digitized data only extends to a spatial frequency of 1.0/(2*pixel size). **res_low** is the lower radius resolution of this range, and it must be equal or larger than **res_hi**. Normally **res_low** can be set to maximum value which means from the canter where radius nearly 0.

res_step is the resolution step size, in another word, it means how many segment should be made in the range from **res_hi** to **res_low**.

C. EXAMPLE PSF INPUT CONTROL PARAMETERS FILE

hv.map.even

hv.map.odd

6.0, 50.0, 50, 2.23

D. PROGRAM EXECUTION

Normal operation of PSF will print out these factors. It is used for comparing two maps (usually the maps generate by odd or even particles). Then we can try to find the effective resolution of our map using the orientation files.

For example:

Execute PSF with 8 nodes using MPI, the command will be:

% mpirun -nolocal -machinefile mach -np 8 PSF < Psf.in > Output

or

% mpirun -np 8 PSF < Psf.in > Result

Psf.in is the script of the input control parameters file, and the **Output** file will be the table of all factors with about **res_step** rows. We use I/O redirect methods. The **mach** file is used for specifying the hosts of the cluster that you want to use. Please modify it to indicate the host names of your cluster. If use second command, it will use the default machine configuration under MPICH directory.

For details of running MPI and specifying hosts in the host file, see user guide of MPICH at http://www-unix.mcs.anl.gov/mpi/mpich/.

E. PROGRAM NOTES

The directories of PSF source code are:

```
|-- Commpk        !common routine directory
|-- PSFsrc        !PSF source codes directory
|-- Vfftpk        !FFT library codes directory
|-- include       !include files directory
|-- Makefile
```

1. In Fourier Domain: annulus = FFT_DIM * pixel_size / resolution

   anuulus <= FFT_DIM / 2

   so: resolution >= 2 * pixel_size

2. FFT transforms used (N is even):

   Analysis: $F(h, k)=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x, y)\exp(-2\pi i(xh+yk)/N)$

   Periodicity: F(h, k) = F(h+N, k) = F(h, k+N) = F(h+N, k+N)

   Conjugate: F*(-h, -k) = F(h, k)

3. factors formula:

   A1 = sqrt(real(F1)**2 + amig(F1)**2)

   A2 = sqrt(real(F2)**2 + amig(F2)**2)

   a. Standard Rfactor

   R-SFactor = Σ|A1-A2| / ΣA1

   b. Rfactor for Phase

R-phase = Σ(sqrt((real(F1)-real(F2))**2 + (aimag(F1)-aimag(F2))**2)) / ΣA1

c. Phase Difference

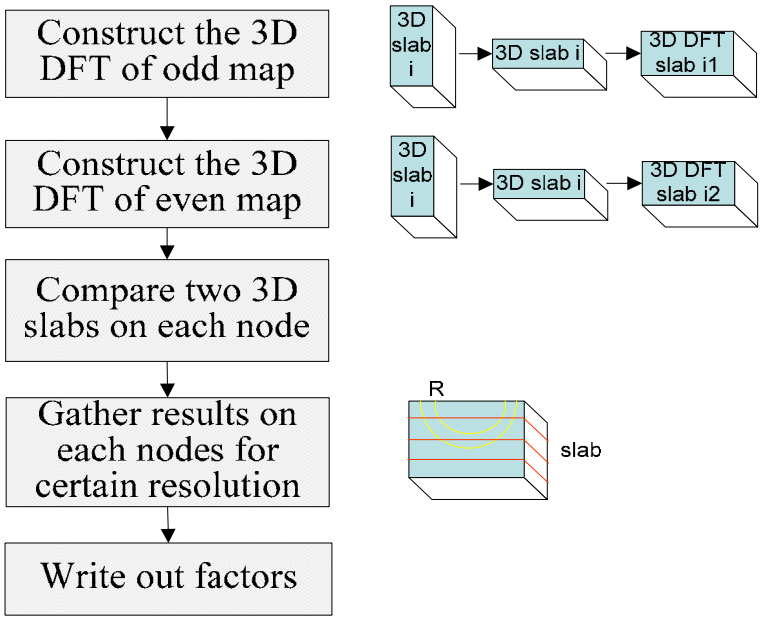
phase-diff = Σ |atan(aimag(F1) / |real(F1)|) - atan(aimag(F2) / |real(F2)|) | / n

d. Correlation Coefficient

$CC = \Sigma((A1 - \overline{A1})(A2 - \overline{A2})) / sqrt(\Sigma((A1 - \overline{A1})^{**}2) * \Sigma((A2 - \overline{A2})^{**}2))$

F. REFERENCES

G. FLOW CHART FOR PSF PROGRAM



PSF !! main program

|--- call usage !! print out usage

|--- call mpi_init !! mpi initialization

|--- call check_map !! check the 1st map

|--- call check_map !! check the 2nd map

|--- call bcast_parameters !! broadcast parameters

|--- call intlz_params !! initialize parameters

|--- call intlz_arrays !! initialize arrays

```
    |--- call vrffti                    !! initialize fft parameter
    |--- for each map do
        |--- call read_map              !! read 3D map and scatter slab to each node
        |--- call fft_2dfft             !! apply 2D DFT onto the 2-slab of 3D model
        |--- call arrange_3d_1          !! rearrange the slab for exchange
        |--- call exch_3d_1             !! exchange slabs among nodes for 1D DFT
        |--- call cfft_1d               !! 1D DFT along z axis with complex number
    |--- call comp_sfactor              !! compute the factors of the two maps
    |--- call mpi_finalize              !! finalize mpi

CHECK_MAP                             !! check the map
    |--- call pif_open                  !! open the pif file for read
    |--- call pif_read_gh               !! read the global header of the pif file
    |--- call pif_read_dh               !! read the data header of the pif file
    |--- call pif_close                 !! close image file

INTLZ_PARAMS                          !! initialize internal parameters
    |--- call trans_length_essl         !! calculate suitable FFT_DIM

INTLZ_ARRAYS                          !! initialize internal arrays
    |--- call set_indices               !! calculate frst_lst_hk
    |--- call set_indices               !! calculate frst_lst_y
    |--- call set_indices               !! calculate frst_lst_u
    |--- call set_indices               !! calculate frst_lst_z

READ_MAP                              !! read and scatter slab of 3D model map
    |--- call pif_open                  !! open the pif file for read
    |--- call pif_read_gh               !! read the global header of the pif file
    |--- call pif_read_dh               !! read the data header of the pif file
    |--- for each slab do
        |--- call pif_read_mapi4        !! read 1 slab of pif file in 4-byte
        |--- call pif_read_mapi2        !! read 1 slab of pif file in 2-byte
    |--- call pif_close                 !! close image file
```