

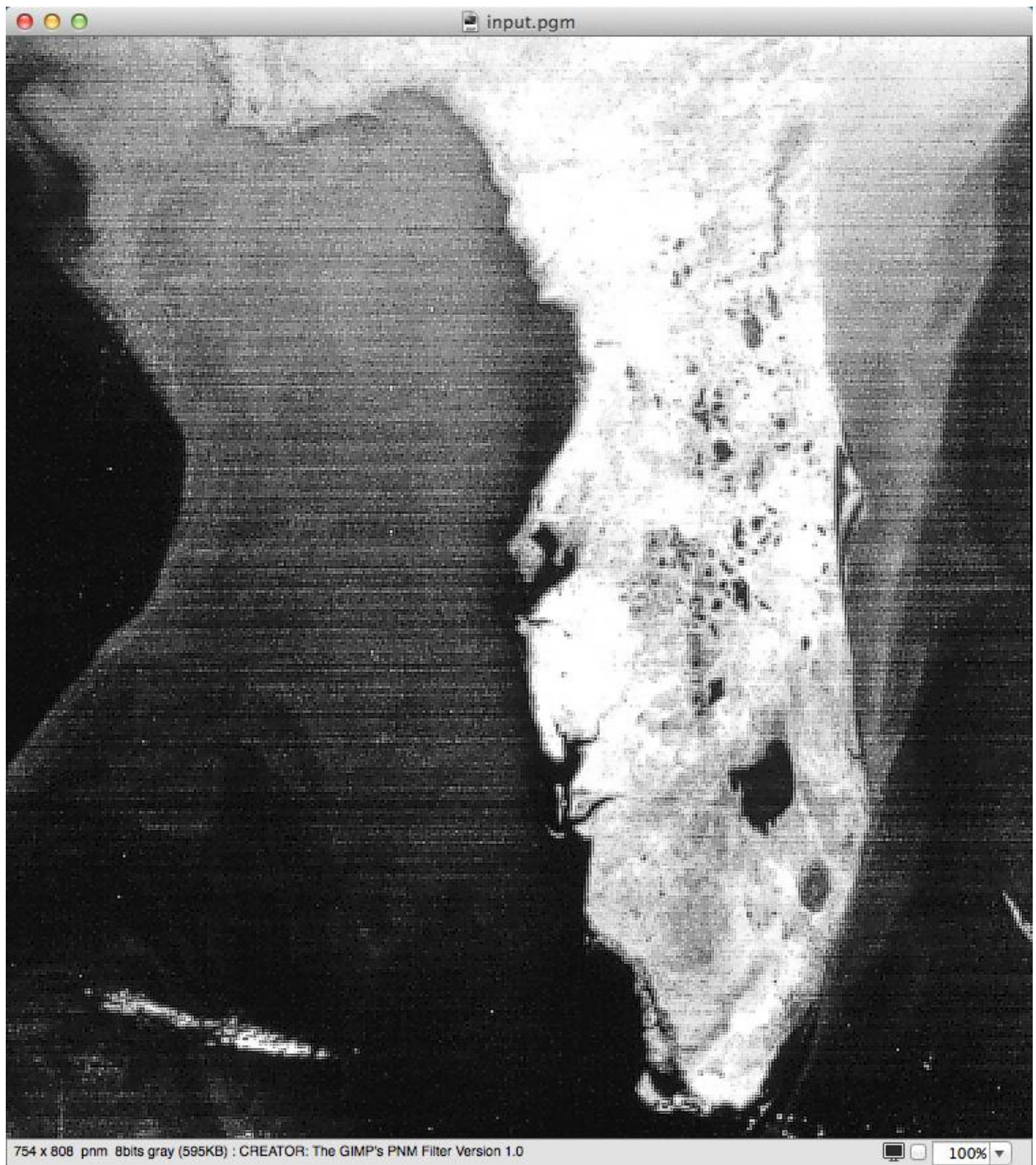
Image Smoothing Using MPI With C

Ye Guo

1. Instruction on how to compile and run my code on cluster
 - 1) Copy the all files in the source folder into cluster.
 - 2) In the file ReadWritePGM.c, the arguments seg1, seg2 and window size could be modified in the head of it. The default values of these arguments are 5, 8 and 20.
 - 3) Modify arguments in file "mpi-run.sbatch", for example, the last line of code in this file, "mpirun -n 2 matrix", you can change "2" into the number of processes plus one like 3 and 4. (The number of processes here should be the number minus 1, which is 2 and 3)
 - 4) Run -sbatch mpi-build.sbatch
 -sbatch mpi-run.sbatch
 - 5) There will be two slurm-XXXX.out (XXXX is the ID of task running on cluster) files generated, the second .out file will show the result of execution.
2. Comparison of running time for different window size and different number of processes

For different window size, the result images are the same, to save some space only paste output image below once!

Source Image:



- 1) winSize = 10
Output Image:



a) $S = 1$

begin reading PGM....

Width=754, Height=808

Maximum=255

The total time cost is 3.296220 seconds

Begin writing PGM....

b) $S = 2$

begin reading PGM....

Width=754, Height=808

Maximum=255

The total time cost is 1.638568 seconds

Begin writing PGM....

c) $S = 4$

begin reading PGM....

Width=754, Height=808

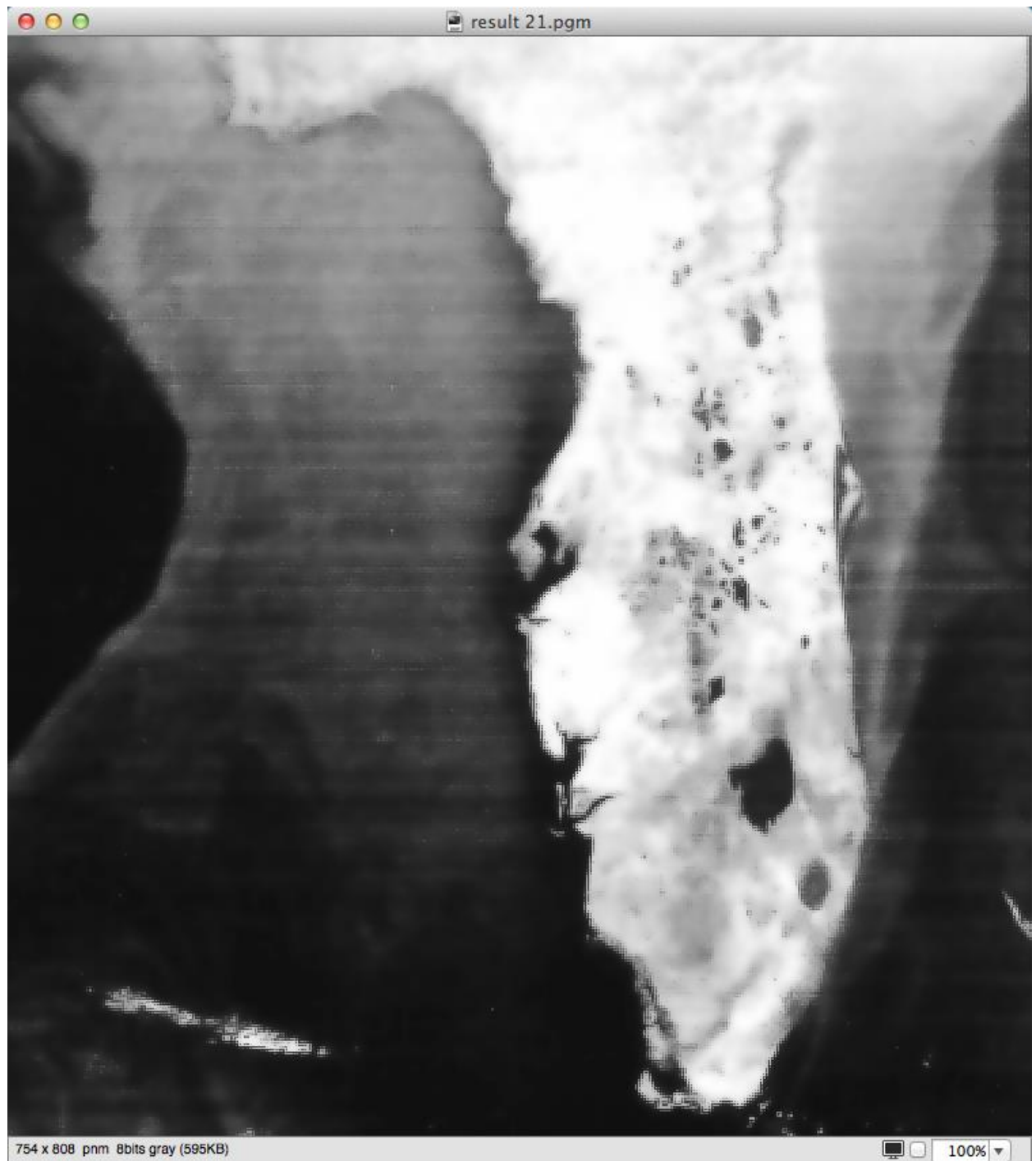
Maximum=255

The total time cost is 1.102386 seconds

Begin writing PGM....

2) winSize = 15

Output Image:



a) $S = 1$
begin reading PGM....
Width=754, Height=808
Maximum=255
The total time cost is 3.853713 seconds
Begin writing PGM....

b) $S = 2$

begin reading PGM....

Width=754, Height=808

Maximum=255

The total time cost is 3.856457 seconds

Begin writing PGM....

c) $S = 4$

begin reading PGM....

Width=754, Height=808

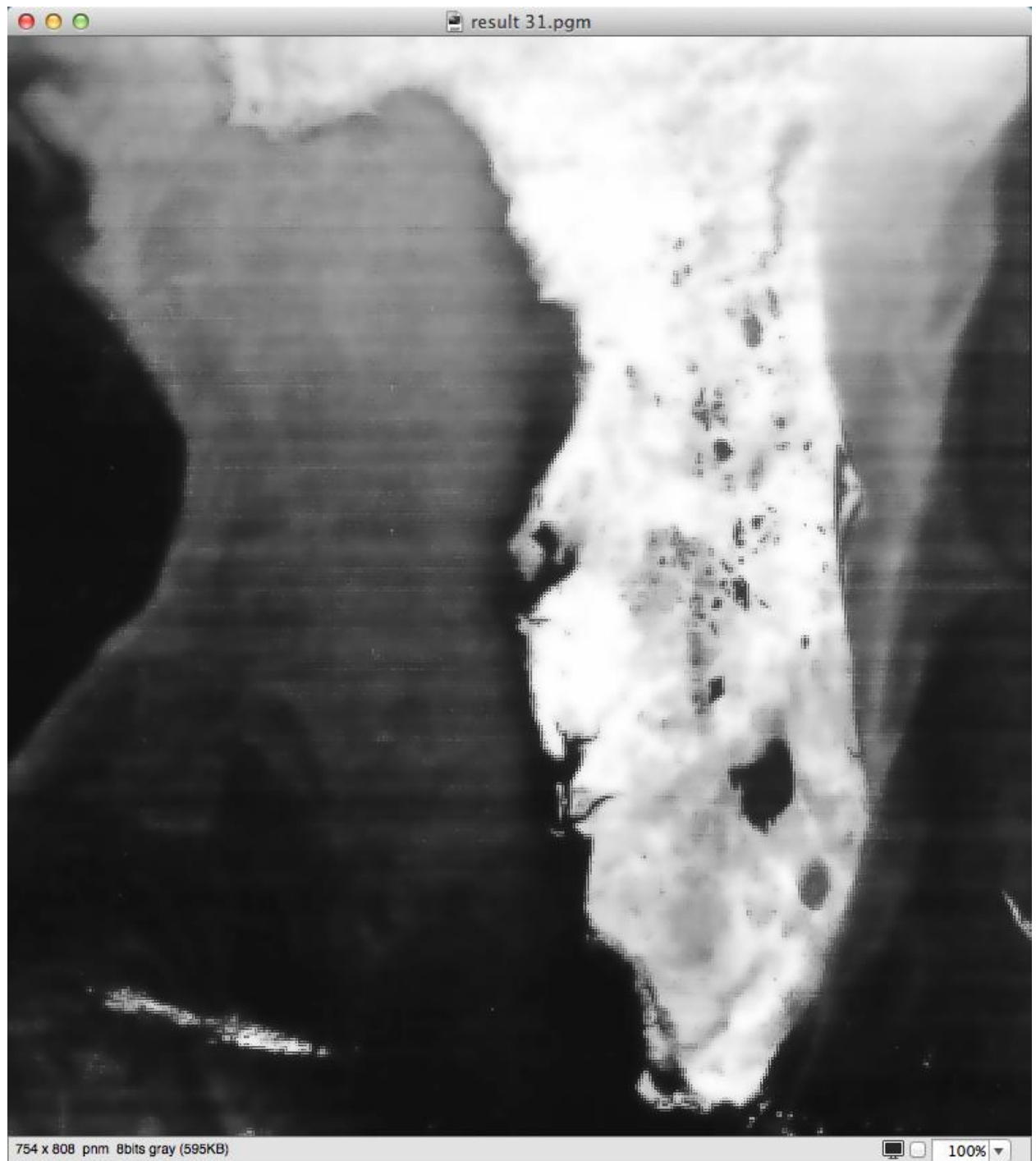
Maximum=255

The total time cost is 2.586898 seconds

Begin writing PGM....

3) winSize = 20

Output Image:



a) $S = 1$

begin reading PGM....

Width=754, Height=808

Maximum=255

[compute-001:02955] 3 more processes have sent help message help-mpi-btl-tcp.txt / invalid if_inexclude

[compute-001:02955] Set MCA parameter "orte_base_help_aggregate" to 0 to see all help / error messages

The total time cost is 7.295146 seconds

Begin writing PGM....

b) $S = 2$

```
begin reading PGM...
Width=754, Height=808
Maximum=255
[compute-001:02963] 3 more processes have sent help message help-mpi-btl-tcp.txt / invalid if_inexclude
[compute-001:02963] Set MCA parameter "orte_base_help_aggregate" to 0 to see all help / error messages
The total time cost is 7.282090 seconds
Begin writing PGM...
```

c) S = 4

```
begin reading PGM...
Width=754, Height=808
Maximum=255
[compute-001:02982] 3 more processes have sent help message help-mpi-btl-tcp.txt / invalid if_inexclude
[compute-001:02982] Set MCA parameter "orte_base_help_aggregate" to 0 to see all help / error messages
The total time cost is 7.323628 seconds
Begin writing PGM...
```

3. Conclusion

- a) The quality of image is decided by arguments seg1, seg2 and also the window size.
- b) The speed of computation is different by choosing different window size and different number of processes used. The more processes and the smaller window size, the faster the execution.