MotionCor2 User Manual

Shawn Zheng
Unviersity of California San Francisco
October 19, 2016

General

MotionCor2, is a multi-GPU program that corrects beam-induced sample motion on dose fractionated movie stacks. It implements a robust iterative alignment algorithm that delivers precise measurement and correction of both global and non-uniform local motions at single pixel level across the whole frame, suitable for both single-particle and tomographic images. MotionCor2 is sufficiently fast to keep up with automated data collection. The result is an exceptionally robust strategy that can work on a wide range of data sets, including those very close to focus or with very short integration times. Application significantly improves Thon ring quality and 3D reconstruction resolution. MotionCor2 is a comprehensive program that integrates gain correction, detection and correction of individual and cluster of bad pixels, dose weighting, and supports both MRC and TIFF file format. MotionCor2 is free for academic use and can be downloaded from **http://msg.ucsf.edu/em/software/index.html**

For technical support, please contact,

Shawn Zheng: szheng@msg.ucsf.edu.

For liscensing MotionCor2, please contact,

David Agard: agard@msg.ucsf.edu
Yifan Chen: YCheng@ucsf.edu

1. System requirement and installation

MotionCor2 is a GPU accelerated program that runs on Linux platform equipped with one or more advanced NVIDIA GPU cards. It has been tested on Tesla k10, k40, k80, GeForce 980, and Titan cards. The current version was compiled on CUDA 7.5. CUFFT library is required. We recommend 128 GB or more CPU memory, although 64 GB CPU memory is also workable.

MotionCor2 is a single-program package. Once unpacked, it is ready to go without any further installation should correct CUDA driver and libraries are installed properly.

2. Quick start

MotionCor2 is a command-line program configurable by means of command-line parameters. The following is the minimum configuration to run the program.

```
MotionCor2 -InMrc /home/data/Stack_0001.mrc \
-OutMrc /home/data/CorrectedSum.mrc
```

In this configuration MotionCor2 corrects only the global motion. It reads in the movie stack "Stack_0001.mrc" and searches the gain reference in the extended header of Stack_0001.mrc. If found, the gain reference is loaded and applied to each frame. If not, the program assumes the stack is already gain corrected. The global motion is then measured and corrected by phase shift in Fourier space.

The minimum configuration for MotionCor2 to correct both global and local motion is as follows.

```
MotionCor2 -InMrc /home/data/Stack_0001.mrc \
-OutMrc /home/data/CorrectedSum.mrc \
-Patch 5 5
```

MotionCor2 first corrects the global motion for each frame and then divides the corrected frames into 5x5 patches on which the local motion is measured. Once completed, the measured shifts from each patch are fitted to a time-varying polynomial model that allows the local motion to be smoothly corrected at single-pixel level aross the whole frame and throughtout the movie stack.

3. Apply gain reference

Gain reference can be saved either in the extended header of MRC files as collected in UcsfImage or UcsfTomo packages or in a separate MRC file as in Leginon. The following example shows how to specify the gain reference in the command line.

```
MotionCor2 -InMrc /home/data/Stack_0001.mrc \
-OutMrc /home/data/CorrectedSum.mrc \
-Gain /home/data/MyGainRef.mrc \
-Patch 5 5
```

In this case MotionCor2 loads the gain reference from "MyGainRef.mrc". If this file is not found, the extended header of "Stack_0001.mrc" will be checked for gain reference. If gain reference is not found, MotionCor2 proceeds without gain correction.

3.1. Rotate and glip gain reference

While the retrieved gain reference is always in accordance with the chip orientation, movie stacks can be collected that are rotated and/or flipped. MotionCor2 allows users to rotate/flip the gain reference to match the orientation of collected movie stacks using command line options -RotGain and -FlipGain.

-RotGain: rotate gain reference counter clockwise. It takes four values from 0 to 3.

0 – no rotation, default,
1 – rotate 90°,
2 – rotate 180°,
3 – rotate 270°.

-FlipGain: flip the gain reference.

0 – no flipping, default,
1 – flip upside down (flip around horizontal axis),
2 – flip left right (flip around vertical axis).

If both -RotGain and -FlipGain are enabled, the gain reference will be rotated first and flipped next.

4. Alignment configuration

Users can configure the number of iterations and the tolerance of alignment accuracy. The corresponding parameters are highlighted in red color in the following example.

```
MotionCor2 -InMrc /home/data/Stack_0001.mrc \
-OutMrc /home/data/CorrectedSum.mrc \
-Gain /home/data/MyGainRef.mrc \
-Patch 5 5 \
-Iter 10
-Tol 0.5
```

In this configuration the iterative alignment procedure terminates when either the alignment error is less than 0.5 pixel or 10 iterations have reached. The default values are 7 for -Iter and 0.5 for -Tol.

5. Discard initial and trailing Frames

The following example shows how to throw away 2 starting frames and 3 frames at the end. The discarded frames are neither included in alignment nor in the corrected sum.

```
MotionCor2 -InMrc /home/data/Stack_0001.mrc \
-OutMrc /home/data/CorrectedSum.mrc \
-Gain /home/data/MyGainRef.mrc \
-Patch 5 5 \
```

-Iter 10 \
-Tol 0.5 \
-Throw 2
-Trunc 3

If not specified, all frames are included.

6. Dose weighting

The following example shows how to enable dose weighting for a single-particle movie stack.

```
MotionCor2 -InMrc /home/data/Stack_0001.mrc \
-OutMrc /home/data/CorrectedSum.mrc \
-Gain /home/data/MyGainRef.mrc \
-Patch 5 5 \
-Iter 10 \
-Tol 0.5 \
-Throw 2 \
-Kv 300 \
-PixSize 0.5 \
-FmDose 1.2 \
```

Users need to specify the high tension in kV, and pixel size of the input stack in angstrom, and the frame dose in e/Å^2. If any of the three parameters is missing, dose weighting step is skipped. When dose weighting is enabled, both dose-weighted and unweighted sums are generated. The weighted sum is saved in the file with its name appended with "_DW" as CorrectedSum_DW.mrc in the aforementioned example. For dose fractionated tomographic tilt series, users also need to specify both the start angle and the tilt step that follows -Tilt.

```
MotionCor2 -InMrc /home/data/Stack_0001.mrc \
-OutMrc /home/data/CorrectedSum.mrc \
-Gain /home/data/MyGainRef.mrc \
-Patch 5 5 \
-Iter 10 \
-Tol 0.5 \
-Throw 2 \
-Kv 300 \
-PixSize 0.5 \
-FmDose 1.2 \
-Tilt 0 2
```

7. Image binning - Fourier cropping

Image binning is implemented by cropping in Fourier domain. -FtBin can be used to bin the motion-corrected image to a specified resolution. Values for this option can be either

an integer or a float that is bigger than 1. In the following case, the output image is cropped in Fourier space by 1.5x.

```
MotionCor2 -InMrc /home/data/Stack_0001.mrc \
-OutMrc /home/data/CorrectedSum.mrc \
-Gain /home/data/MyGainRef.mrc \
-Patch 5 5 \
-Iter 10 \
-Tol 0.5 \
-Throw 2 \
-FtBin 1.5
```

If the raw movie stacks are collected in super-resolution mode and the final images is intended to be binned, we recommend to use the super-resolution stacks as input and let MotionCor2 do the binning. This is a better practice than passing binned stack to MotionCor2. Since the local motion is corrected by linear interpolation that has low-pass effect in Fourier space, it is preferred to correct the local motion at super-resolution pixel to minimize the loss of high-frequency information due to interpolation.

8. Run on multiple GPUs

The following example demonstrates how to use multiple GPUs.

```
MotionCor2 -InMrc /home/data/Stack_0001.mrc \
-OutMrc /home/data/CorrectedSum.mrc \
-Gain /home/data/MyGainRef.mrc \
-Patch 5 5 \
-Iter 10 \
-Tol 0.5 \
-Throw 2 \
-Kv 300 \
-PixSize 0.5 \
-FmDose 1.2 \
-FtBin 2 \
-Gpu 0 1 2
```

9. Batch processing

Batch processing overlaps the disk operation with the intensive computation involved in motion correction such that the disk I/O time is almost completely shadowed by the computational time. MotionCor2 automates the sequential motion correction of multiple single-particle movie stacks based upon pattern recognition of file names. "-Serial 1" enables the batch processing. There are two scenarios. First, when the folder contains only the movie stacks, the following examples shows the how to configure the command line.

```
MotionCor2 -InMrc /home/data/ \
-OutMrc /home/Sum/Corrected \
```

```
-Gain /home/Ref/MyGainRef.mrc \
-Patch 5 5 \
-Iter 10 \
-Tol 0.5 \
-Throw 2 \
-Kv 300 \
-PixSize 0.5 \
-FmDose 1.2 \
-FtBin 2 \
-Gpu 0 1 \
-Serial 1
```

In this case all the MRC files in "/home/data/" are treated as movie stacks and corrected sequentially. The corrected sums are named by prefixing "Corrected" to the input file names and saved in "/home/Sum/" directory.

Second, if the input folder contains mixed files, the following example shows how to configure the command line for the program to choose only the MRC stack files.

```
MotionCor2 -InMrc /home/data/Stack_ \
-InSuffix Raw.mrc \
-OutMrc /home/Sum/Corrected \
-Gain /home/Ref/MyGainRef.mrc \
-Patch 5 5 \
-Iter 10 \
-Tol 0.5 \
-Throw 2 \
-Kv 300 \
-PixSize 0.5 \
-FmDose 1.2 \
-FtBin 2 \
-Gpu 0 1 \
-Serial 1
```

MotionCor2 chooses only the files with names prefixed with "Stack_" and suffixed with "Raw.mrc" in "/home/data" directory. Here are two examples, "Stack_1234Raw.mrc" and "Stack_3456-Raw.mrc".

10. Support for TIFF files

-InTiff is used specify an input of TIFF file. Currently, there is no support for batch processing of TIFF files. The support of TIFF files is limited to single-particle movie stacks.

```
MotionCor2 -InTiff /home/data/Stack_0001.tif \
-OutMrc /home/Sum/Corrected_0001.mrc \
-Gain /home/Ref/MyGainRef.mrc \
```

-Patch 5 5 \
-Iter 10 \
-Tol 0.5 \
-Throw 2 \
-Kv 300 \
-PixlSize 0.5 \
-FmDose 1.2 \
-Gpu 0 1

11. Output motion corrected stack

Motion corrected stacks can be generated by specifying "-OutStack 1" along with the motion corrected sum. Note that in this setup the dose weighting step is skipped. As such, the corrected sum and stack are not dose weighted. The output stack is stored in a MRC file with "_Stk" appended to the end of the output file name. This option is not available for dose fractionated tomographic tilt series.

12. Low-signal movie stacks

There are two parameters users can play with. The first is B-factor default to 1. Its value can be changed by -bft.

Another approach is to adjust the setting of -Group with its value default to 1. For stacks with low signal to noise ratio, using a higher value has been found very effective. -Group instructs the program to equally divide the input stack into sub-groups. Instead of aligning individual frames, the sums these sub-groups are aligned. The shifts of individual frames are then interpolated and extrapolated. For example, -Group 3 divides the input stack into consecutive (non-overlapping) sub-groups, each containing 3 frames. As opposed to increasing B-factor, this is a recommended approach.

13. Miscellaneous

Starting the program without specifying any argument will display all the command line parameters with brief discriptions.

Release Report

10-19-2016 version:

1. Bug fix in generation of log file.
2. Bug fix in saving MRC file. 1) Add min, max, mean in the main header. 2) Revise the main header in accordance to the format defined in JSB 2015, 192 (2) 146-150.
3. Bug fix in the determination of whether the input MRC file is a tomographic tilt series or a single-particle movie stack.

Frequently Asked Questions

1. The input movie stack is already gain corrected. MotionCor2 reports on the terminal "Apply gain to the stack". Will the gain reference be applied again in MotionCor2?
   No, as long as the gain reference is not provided from the command line as a MRC file or not contained in the extended header of the MRC file of the input movie stack.

2. Are the bad and hot pixels detected different from camera_defects_pixels?
   Yes, they are different since MotionCor2 detects them dynamically.

3. I got the following error messages, what went wrong?
   "Error: CCufft2D failed, unable to create CUFFT_R2C plan."
   "Error: CCufft2D::Forward: an illegal memory access was encountered."
   "Error: CCufft2D::Forward: an illegal memory access was encountered."
   These error messages are most likely caused by incompatiblity between CUDA driver and CUDA toolkit. Ask system administrator to check if both the driver and the toolkit are of the same version.

4. I noticed that the output and log file always list the frame shifts relative the first frame, no ma matter what value "-FmRef" is set to.
   The output and log files list the shifts relative to the first frame. However the correction is relative to the central frame by default. "-FmRef" is a switch that allows to choose the reference either the central frame by giving it a non-zero value or the first frame by setting it zero.

5. Does MotionCor2 work on Falcon images?
   Yes.