MotionCor2 User Manual

Shawn Zheng
Unviersity of California San Francisco

Version: 1.1.0
Release Date: 05/03/2018

General

MotionCor2 is a multi-GPU program that corrects beam-induced sample motion recorded on dose fractionated movie stacks. It implements a robust and efficient iterative alignment algorithm that delivers precise measurement and correction of both global and local motions at single pixel level, 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**

Contacts:

1. Suggestions, discussions, and technical support:
   Shawn Zheng: szheng@msg.ucsf.edu

2. Liscensing MotionCor2:
   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 is tested on many current nVidia GPU cards. The current version was compiled on Centos 7. Three executables are provided for CUDA 8.0, 9.0, and 9.1, respectively. CUFFT library is required. We recommend 128 GB or more CPU memory, although 64 GB CPU memory is also workable depending on the size of movie stacks.

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

MotionCor2 can process movies stacks that are saved either in MRC or TIFF files. The following are example-based explanations for running MotionCor2.

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 for 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 across the whole frame and throughout the movie stack.

2.1. Overlapping between adjacent patches

"-Patch" can have an optional third parameter to specify the overlapping between adjacent patches. For example, "-Patch 5 5 20" means that each patch will have 20% overlapping with its neighboring patches in each dimension.

3. Running on multiple GPUs

MotionCor2 can run on multiple GPUs by distributing computation onto each participating GPU. The faster GPU will be assigned more computation than those slower ones. The following example shows the configuration of using 4 GPUs.

```
MotionCor2 -InMrc /home/data/Stack_0001.mrc \
-OutMrc /home/data/CorrectedSum.mrc \
-Patch 5 5 \
-Gpu 0 1 2 3
```

In this example the hosting node has 4 GPUs installed with unique IDs of 0 1 2 3. The GPU IDs can be found by running a nVidia program nvidia-smi on command line. GPU 0 will be used if -Gpu is not present in the command line.

It is very important to note that from version 1.0.1 each GPU can only be used in one MotionCor2 process. If any GPU is shared in two or more MotionCor2 processes, motion correction may fail! This is because the GPU memory allocated in MotionCor2 is not managed and the allocation cannot exceed the physical memory each GPU has. It is strongly recommended to use nvidia-smi to check before starting MotionCor2. The following is an example of results reported by nvidia-smi.

```
nvgpu-3-8 101% nvidia-smi

+-----------------------------------------------------------------------------+
| NVIDIA-SMI 367.48                 Driver Version: 367.48                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GeForce GTX 1080    On   | 0000:03:00.0     Off |                  N/A |
| 13%   53C    P2    33W / 215W |    157MiB /  8113MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  GeForce GTX 1080    On   | 0000:84:00.0     Off |                  N/A |
| 16%   55C    P2    49W / 215W |    157MiB /  8113MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|    0     15716    C   MotionCor2                                     155MiB |
|    1     16045    C   MotionCor2                                     155MiB |
+-----------------------------------------------------------------------------+
```

Fig. 1 nvidia-smi shows that two GPUs, 0 and 1, are installed. Both of them are currently used by a MotionCor2 process.

3.1. GPU memory usage

In case multiple MotionCor2 processes are needed to run side by side, -GpuMemUsage is recommended to use. By default, it is set to 0.5, meaning 50% of GPU memory is allocated to buffer the movie stacks. If not enough, CPU memory is used to buffer the remaining frames. It will be safe to set "-GpuMemUsage 0" when multiple processes are needed.

4. 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.

4.1. 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.

5. Dark reference subtraction

Dark reference can be loaded from a MRC file. If it is loaded successfully, dark reference will be subtracted from each frame before gain reference is applied. The following command line shows how to specify dark reference in the command line.

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

***** Here we assume dark and gain references bear the same orientation. This means that -RotGain and -FlipGain settings are also applied to the dark reference if they are specified in the command line. *****

6. 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 \
-Dark /home/data/MyDarkRef.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.

7. 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 \
-Dark /home/data/MyDarkRef.mrc \

-Patch 5 5 \
-Iter 10 \
-Tol 0.5 \
```

-Throw 2
-Trunc 3

If not specified, all frames are included.

8. Align to specified frames

By default a movie stack is aligned to its central frame. However, there are some occasions when users want to align their stacks to a specified frame. This is can be done by specifying the frame number after –FmRef. The frame number is zero-indexed based upon only the loaded frames.

9. Dose weighting

MotionCor2 implemented the dose weighting scheme developed by Grant et al. [1]. 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

10. Correct anisotropic magnification

Anisotropic magnification causes images less magnified in one direction (major axis) and more magnified in the direction (minor axis) perpendicular to major axis. MotionCor2 corrects anisotropic magnification by stretching the image along the major axis. Users need to obtain the parameters of anisotropic magnification using Tim Grant's program **mag_distortion_estimate**[2]. These parameters can then be provided to MotionCor2 using "-Mag" that is followed by major scale, minor scale, and the angle of the major scale. In the following example MotionCor2 corrects the anisotropic magnification that has major scale of 1.003, minor scale of 0.998, and 34.0° of distortion angle reported by mag_distortion_estimate.

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 \
-Mag 1.003 0.998 34.0

Note: In the previous versions of MotionCor2, "-Mag" should be followed by magnifications rather than scales. Therefore, users need to invert the scales to magnifications.

11. 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.

12. 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".

13. 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

13.1. Batch processing of TIFF stacks

MotionCor2 now supports batch processing of TIFF stacks in the same fashion as it does for MRC stacks except that "-InTiff" should be used instead and "-InSuffix" should be followed with the corrrect file extension of TIFF stacks.

14. 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.

15. Low-signal movie stacks

There are two parameters users can play with. The first is B-factor default to 100. Its value can be changed by -Bft. Since version 1.1.0, -Bft takes two parameters of which the first one is used in global-motion measurement and the second is for local-motion. The default values are "-Bft 500 150".

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.

16. Correction of defects on camera

In addition to dynamically detect and correct defects in acquired movie stacks, users can specify fixed regions of defects in a text file. This file is composed of multiple lines of which each contains four space-separated integers, x, y, w, and h that define a rectangular region of defects. x and y are the pixel coordinates of the lower left corner of such a region where w and h denote the width and height, respectively. The full path of this text file should follow the tag "-DefectFile". The defective pixels will be replaced with random picks of good pixels in their neighborhood.

17. Archive raw movie stacks

This function allows to archive movie stacks in MRC files with each pixel saved in 4 bits. This function can only be enabled when (1) the input movie stack is given in MRC file, (2) the MRC file has 8 bits per pixel corresponding to MRC mode of 0 or 5, and (3) the full path of the archive file is provided behind "-ArcDir".

If the gain reference is provided as a MRC file in the command line, the gain reference will be saved at the end of the extended-header area in the archive file.

18. Taking into account of frame motion blurring

Since version 1.1.0, there is a newly introduced option "-InFmMotion 1" that takes into account of motion-induced blurring of each frame. The test on T20S proteasome data set

shows, although not significant, noticeable resolution improvement of reconstruction. By default, this option is off.

19. Miscellaneous

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

Release Report

11-30-2016 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.

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.

01-03-2017 version:

1. Add function to correct anisotropic magnification.
2. Add –FmRef that allows movie stacks to be aligned to a user-specified frame.

Version 1.0.0 – Release on 07-05-2017

Per user requests, we start using version number instead of date to track the release of MotionCor2. The first version number indicates major changes have been made. The second number denotes minor changes or features added. The last number typically refers to bug fixing.

1. Allow users to specify camera defects in an input file.
2. Support batch processing of TIFF stacks.
3. Support archiving MRC stacks.
4. Revised how users should enter the parameters of anisotropic magnification.
5. Provide "-PhaseOnly" option for cross correlation used in alignment.

Version 1.0.1 – Release on 09-06-2017

This version significantly improves computational efficiency by buffering as many frames in GPU memory as possible. This strategy significantly reduces the overhead of copying frames from CPU to GPU. As a result, as much as 50% of the computational time can be saved.

Importantly, each GPU cannot be shared by two MotionCor2 processes at the same time. If a GPU is used in one MotionCor2 process, the second MotionCor2 process should not use this GPU. Otherwise, both processes will yield incorrect results and CUDA errors due to GPU memory limitation.

The new functions are:
1. Dark-reference correction.
2. Overlapped patches for local motion correction.

Version 1.1.0 – Release on 09-06-2017

This version further improves the speed of motion correction and more robust compared to previous releases.

The new functions are:
1. -GpuMemUsage allows users to specify how much GPU memory is used to buffer movie frames.
2. -InFmMotion allows to take into account motion-induced blurring on each frame.
3. -Bft takes an optional second parameter that is used for measuring local motion.

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.