

/specs description

Tuesday, February 28, 2023 6:08 PM

In the description `specs` is an object of a class `MovieSpecs`. `specs/extra_specs` aren't required, but sometimes used and have some interfaces baked in.

As discussed, here is my suggestion on the file content (in addition to the movie) that would be convenient to have in every .h5 file (conversion from dcimg to h5 adds this content and all subsequent functions update it accordingly). I use `MovieSpecs` class for that. As an example, I added an option in `convertRaw2Preproc1` that allows to call it with this approach (options.useMovieSpecs, the default behavior should be unchanged, see commit [78849b6](#)). I think I personally will be using this approach (I'm flexible in details in case you want too). `[movie, movie_specs] = rw.h5readMovie(h5filename, varargin)` and `movie_specs = rw.h5readMovieSpecs(h5filename)` in are interfaces to read the specs structure from .h5 file

From <https://github.com/schnitzer-lab/VoltageImagingAnalysis/issues/98>

Name	Description	Ways to interact	Reason	Example
<code>/fps</code>	float. Camera fps for the initial recording	Set automatically when the movie is converted to .h5 to and averaged fps during the recording (from timestamps), to get the value use <code>specs.getFps()</code> , it will also account for <code>timebinning</code> property	To be sure what is the absolute timing for the recording in the current file	100.4
<code>/timebinnig</code>	int(?). Temporal downsampling (relative to camera recording)	To add binning use <code>specs.AddBinningTime(n)</code> .	Combined with <code>/timeorigin</code> provides a way to track the time position if the part of the recording in the current file relative to original recording	1
<code>/timeorigin</code>	int. First frame number (in the raw camera recording frames numbering) of the part of the movie in the current file	Set automatically to the first converted frame when the movie is converted to .h5. To add delay use <code>specs.AddFrameDelay(nframes)</code> .	Combined with <code>/timebinnig</code> provides a way to track the time position if the part of the recording in the current file relative to original recording	3000
<code>/pixsize</code>	float. Size of current pixel in mm.	Set automatically to $8/2048 = 0.0039$ mm (BFM FOV size over the full camera resolution) when the movie is converted to .h5, to get the value use <code>specs.getPixSize()</code> , it will also account for <code>binning</code> property	Knowledge of the absolute spatial scale	0.0039
<code>/binning</code>	float(?). Spatial downsampling (relative to raw camera recording).	Set automatically to (software binning) x (hardware binning) during the conversion to .h5, can be changed using <code>specs.AddBinning(n)</code> .	Combined with <code>/spaceorigin</code> provides a way to track the position of the part of the FOV in the current file relative to original recording; likely be useful for brain region alignment.	8,8
<code>/spaceorigin</code>	[int, int]. Origin (up-left corner) of the part of the FOV in the current file in the raw camera recording pixel coordinates.	To add cropping use <code>specs.AddSpatialCropping(p)</code> . To get values appropriate for the current binning, use <code>specs.getSpaceOrigin()</code> .	Combined with <code>/spacebinning</code> provides a way to track the position of the part of the FOV in the current file relative to original recording; likely useful for brain region alignment.	0,0
<code>/sourcePath</code>	char array. Path to the original .dcimg recording	Set automatically when the movie is converted to .h5	To know where the data comes from	'Q:\GEVI_Wave\Raw\Anesthesia\m11_20201013\meas02\m11_d201013_s02_--fps118-cG.dcmg'
<code>/history</code>	Char array; string with a fixed separator (;) that lists major processing steps that happened the file	To add processing step, use <code>specs.AddToHistory(new_entry)</code>	Makes the processing steps tractable for every file, no need for complicated, unclear and inconsistent file naming	'convertRaw2Preproc1;regMovies;moc oMovies'
<code>/history_params</code>	Char array; json-formatted string where every <code>/history</code> step has a corresponding entry with all the parameters of the high-level (file) function call.	Added together with a processing step <code>specs.AddToHistory(functionCallStruct({'param1', 'param2', 'options'}))</code> (or <code>specs.AddToHistory(new_entry, params)</code>). For back compatibility, if the field is absent in the file read, it will be added with <code>params_not_saved = true</code> and no parameters listed.	To be able to recover all the processing steps	'[{"001:convertRaw2Preproc1":{}}, {"002:regMovies":{}}]'
<code>/extra_specs</code>	A Map container; a name-value pare, all the specs that are not required to be in the file			
<code>/extra_specs/allenMapEdgeOutline</code>		See <code>pipeline_allen_align.m</code> and <code>MovieSpecs.getAllenOutlines()</code>		
<code>/extra_specs/mask</code>		See <code>pipeline_mask_add.m</code> and <code>MovieSpecs.getMask()</code>		
<code>/extra_specs/timestamps_table</code>		See <code>MovieSpecs.getTTLTrace()</code>		
<code>/extra_specs/frange_valid</code>	What is the frequency range of the time signal in the file	See <code>MovieSpecs.AddFrequencyRange()</code> and <code>MovieSpecs.getFrequencyRange()</code>		

file (due to filtering)	MovieSpecs.getFrequencyRange()	
-------------------------	--------------------------------	--