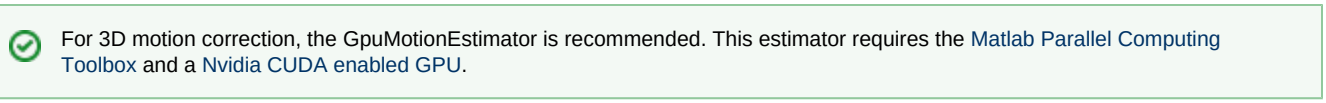
Motion Correction

Introduction

ScanImage can continuously detect XYZ motion of the currently acquired image relative to a reference volume during an active acquisition.

The motion correction can be used for

- Detect XYZ motion during a volume acquisition
- Use multiple ROIs as motion correction reference
- Correct for motion by moving actuators
- Retargeting the stimulation laser during a Photostimulation experiment
- Tracking image features for online analysis with ScanImage's ROI Integration feature
- Finding the field of view from a previous session



Setup

Collect a reference stack. Right click on the volume in the channel view window and select 'Set as Motion Correction Reference'.

Motion Estimators

ScanImage ships with 3 Motion Estimators. All estimators uses basic slice-wise phase correlation to find the best match between the acquired slice and the reference volume.

Name	System Requirements	Performance	Description
SimpleMotionEstimator	None	Good	Requires no additional toolboxes. Not well suited for 3D motion correction due to performance issues.
GpuMotionEstimator	▪ Parallel Computing Toolbox ▪ Nvidia CUDA enabled GPU	Best	Best suited for 3D Motion Correction. Note: processing data on the GPU is fast, but transferring data to the GPU is a bottleneck. When imaging with low resolution, the SimpleMotionEstimator might perform better.
ParallelMotionEstimator	• Parallel Computing Toolbox	Better	Alternative to the GPUMotionEstimator if no GPU is present. This estimator uses parallel workers for processing and does not slow down the acquisition. The tasks are queued for processing. The queue size is a user settable property.

Motion Correctors

ScanImage ships with 1 Motion Corrector.

Name	Description
SimpleMotionCorrector	This motion corrector averages the motion estimates of the last N seconds. If average motion vector is greater than the correction threshold, a correction event is triggered. The minimum time in between correction events is settable by the property correctionInterval_s.

API

Motion Estimators

Motion estimators derive from the class scanimage.interfaces.IMotionEstimator

The reference volume and the image data are handed to the Motion Estimator as instance of the class scanimage.mroi.RoiData.

scanimage.mroi.RoiData contains information about the ROI geometry (hRoi), the channels (channels) and the currently imaged zs (zs). The image data is stored in the property imageData. imageData is a cell array, where the first index is the channelIdx, and the second index is the z index.

The function

motion_estimator_result = estimateMotion(obj,roiData)

does not return the motion estimate directly, but instead returns an object of type scanimage.interfaces.IMotionEstimatorResult. ScanImage then polls this class obtain the estimation results. The purpose of this class is to enable asynchronous processing.

Motion Correctors

Motion estimators derive from the class scanimage.interfaces.IMotionCorrector

When a new motion estimate is available, scanimage populates the estimate by calling the function updateMotionHistory. This hands the entire motion history to the corrector. The corrector can then analyze the history and determine if a correction is required. When the corrector wants to initiate a correction, it notifies its event 'correctNow'. ScanImage then queries the function 'getCorrection' to get the correction value. Note: if the corrector returns an invalid value (e.g. values outside the allowable correction range), ScanImage discards the correction event. After a correction is performed, ScanImage calls the function correctedMotion.