Motion Correction

Introduction

ScanImage can continuously detect X-Y motion of the currently acquired image relative to a reference image during an active acquisition. ScanImage can also estimate motion in Z by comparing the currently acquired image to a reference volume of images during an active acquisition.

The motion correction can be used for

- Retargeting the stimulation laser during a Photostimulation experiment
- Tracking image freatures for online analysis with ScanImage's ROI Integration feature
- Finding the field of view from a previous session
- Stabilizing the image display
- Maintain scan position in Z to correct for drift

Setup

Algorithm

ScanImage ships with five algorithms for motion detection:

ScanImage Default	Function	Performance	Description
Default if GPU acceleration is not available	fftCorr	Good	estimates motion by calculating the fft-based circular cross correlation between the reference image and the current image
Default if GPU acceleration is available	fftCorrGpu	Excellent	same as fftCorr, but calculates the fft-based cross correlation using the GPU
	fftCorrSideProj	Excellent	first calculates the mean side projections for x and y, then calculates the fft-based circular cross correlation between the side projections of the reference image and the current image; less accurate, but more performant than fftCorr
	fftPhaseCorr	Variable	estimates the approximate Z position by calculating the fft-based normalized phase cross correlation between the current image and all images in a reference volume, after determining which plane in the reference volume best matched the current image the X-Y offset is calculated between the current image and reference in the same manner as above. If the reference does not contain multiple images then this behaves the same way as fftCorr. Performance varies depending on size and resolution of reference volume
	fftPhaseCorrGpu	Good	same as fftPhaseCorr but uses the GPU

GPU acceleration for motion correction is only available if the Matlab Parallel Computing Toolbox is installed and a CUDA enabled GPU is available.

The motion correction algorithm is split into two parts:

- Reference image pre-process function
- Motion correction algorithm

Use the Motion Correction Windows to select which motion correction algorithm is used.

The Motion Correction Functions are stored in +scanimage\+components\+motionCorrection
When selecting a motion correction function in the Motion Correction Window, make sure to also select the corresponding preprocess function.

API

The pre-process function is called only when a new reference image is loaded, while the motion correction algorithm is called every time a frame is acquired. This reduces the number of reoccuring calculations. In the fft-based cross correlation, the fft of the reference image is processed once, and then stored for later use. The preprocess function ha two return values:

Return Value	Description
refDisplayImage	(Required) Used as a reference image in the Motion Correction Display Windows
refImagePreProcessed	(Required) The pre-processed reference data passed to the motion correction algorithm whenever a frame is acquired. (Can be a matrix, struct, class..)

The motion-correction function expects two inputs:

Input	Description
refImagePreProcessed	pre-processed reference data
image	image data of current frame

Output	Required	Data type	Description
success	Required	logical	Indicates if the motion algorithm found a match between the reference image and the current image
ijkOffset	Required if success==true	[1x3] double	The [i,j,k] offset between the reference image and the current image. The k offset is actually the slice number of best match, actual Z position is then determined by the slice number and the step per slice.
quality	Can be empty	double	A metric describing the quality/certainty of the match
cii	Can be empty	[1xM] double	A metric showing the quality/certainty of the match for each possible pixelshift in direction i
cjj	Can be empty	[1xN] double	A metric showing the quality/certainty of the match for each possible pixelshift in direction j

Note: ScanImage internally represents images in column-major order, whereas Matlab uses row-major order for storing matrices. This means that images appear to be transposed when plotted.
As a workaround, images can be transposed, which will result in a performance hit. Alternatively, the axes view can be rotated:

```
view(gca,-90,90);
```

The motion correction functionality is handled by the ScanImage module hMotionManager. Type

```
hSI.hMotionManager
```

in the Matlab Command Window to see all availble properties and functions.