Robust non-rigid alignment of volumetric calcium imaging data

Benjamin Poole, Logan Grosenick, Michael Broxton, Karl Deisseroth, Surya Ganguli

Abstract: Improvements in genetically-encoded calcium indicators have led to ever broader application of calcium imaging as a means to record population activity with cellular resolution in targeted neural populations. However, during *in vivo* experiments, brain motion relative to the sensor results in motion artifacts both within and between images. Such artifacts render it difficult to accurately estimate calcium transients using fixed spatial regions of interest, making image alignment a critical first step in analyzing calcium imaging datasets.

Previous alignment algorithms for calcium imaging require the specification of a fixed template image that can be matched to each frame. These algorithms are ineffective when applied to the latest generation of optical activity reporters, which have both negligible background signal and large changes in neural activity over time. Our work is based on RASL, an image alignment technique that learns a low-rank matrix that represents a set of templates that can be adaptively combined to match each frame, and a sparse matrix that represents deviations from the adaptive template (Peng et al., 2010). This allows us to leverage low-dimensional dynamics in neural activity to robustly align functional data without a fixed template.

We extend RASL to account for three-dimensional translations, rotations, and non-rigid deformations such as those caused by scanning artifacts. To scale RASL to datasets containing millions of voxels and thousands of frames, we introduce several extensions including randomized decompositions and online alignment. We validate our technique on images and volumes from two-photon and light field microscopy (Grosenick et al., 2009; Broxton et al., 2013), showing improved accuracy and reduction of motion artifacts compared with existing techniques. An implementation of our algorithm is released as a Python package using the Apache Spark distributed computing framework and integrated with Thunder (Freeman et al., 2014).