A hardware system for real-time decoding of *in-vivo* calcium imaging data

Zhe Chen[1,2], Garrett J. Blair[2], Changliang Guo[3,4], Alicia Izquierdo[2,5], Peyman Golshani[3,4,5], Jason Cong[1], Daniel Aharoni[3,4,5], Hugh T. Blair[2,5]

[1]Department of Electrical and Computer Engineering, UCLA, Los Angeles, CA 90095-1563, USA.

[2]Department of Psychology, UCLA, Los Angeles, CA 90095-1563, USA.

[3]David Geffen School of Medicine, University of California Los Angeles, Los Angeles, CA, 90095 USA

[4]Department of Neurology, David Geffen School of Medicine, University of California, Los Angeles, Los Angeles, CA, USA

[5]Integrative Center for Learning and Memory, University of California, Los Angeles, Los Angeles, CA, USA

ABSTRACT

We present a low-cost, open source hardware platform for online decoding of *in vivo* calcium signals with the UCLA Miniscope. Device performance was evaluated by decoding hippocampal population activity in unrestrained rats as they ran on a linear track. We show that the system can accurately decode a rat's position on the track in real time, and is capable of triggering closed-loop feedback at millisecond latencies after detecting fluorescence patterns.

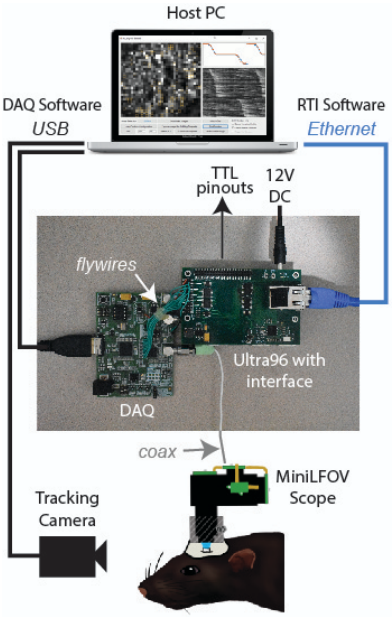


Figure 1. *System Hardware*. Miniscope connects to DAQ via coax cable, DAQ connects to Ultra96 via flywires, host PC connects to Ultra96 via Ethernet and to DAQ via USB 3.0; TTL pinouts from Ultra96 can drive closed-loop feedback stimulation from external devices.

MAIN

Epifluorescence miniature microscopes ("miniscopes") can be worn on the head of a freely-behaving animal to perform *in vivo* calcium imaging of neural population activity[1,2,3,4,5]. Imaging data is usually collected while subjects are engaged in a task, and stored for later offline analysis. Emerging techniques for online image analysis[6,7,8,9,10,11] offer the potential for novel real-time experiments in which active interventions (such as neurostimulation or sensory feedback) are triggered immediately in response to neural activity decoded from calcium fluorescence[12,13,14]. But to our knowledge, a functional system for real-time population decoding of calcium traces is not yet available to the research community.

Here, we introduce a low-cost hardware platform for online population decoding of *in vivo* calcium signals that can generate closed-loop feedback at millisecond latencies. The system receives real-time video input from the UCLA Miniscope data acquisition (DAQ) interface, and is thus compatible with several miniscopes that use this interface[3,15,16,17]. Results below were obtained using a Large Field-of-View version of the UCLA Miniscope (MiniLFOV)[17] featuring 1296x972 pixel resolution acquisition (cropped to 512x512 for real-time processing; see Extended Data Fig. 1) at 22.8 frames-per-second (fps). The MiniLFOV was connected via coax cable to a modified Miniscope data acquisition (DAQ) board (Fig. 1), with 14 flywires soldered to the PCB for transmitting video data to an Avnet™ Ultra96 development board where online image processing (motion correction, image enhancement, calcium trace extraction, and decoding) was performed by our ACTEV (Accelerator for Calcium Trace Extraction from Video) firmware running on a Xilinx™ Zynq Ultrascale+ MPSoC. The system was controlled from a host PC running standard Miniscope DAQ software (to focus and adjust the MiniLFOV) and newly developed real-time interface (RTI) software to communicate with ACTEV.

Device performance was evaluated using image data collected from the hippocampal CA1 region while Long-Evans rats ran back and forth on a 250 cm linear track (Fig. 2A). CA1

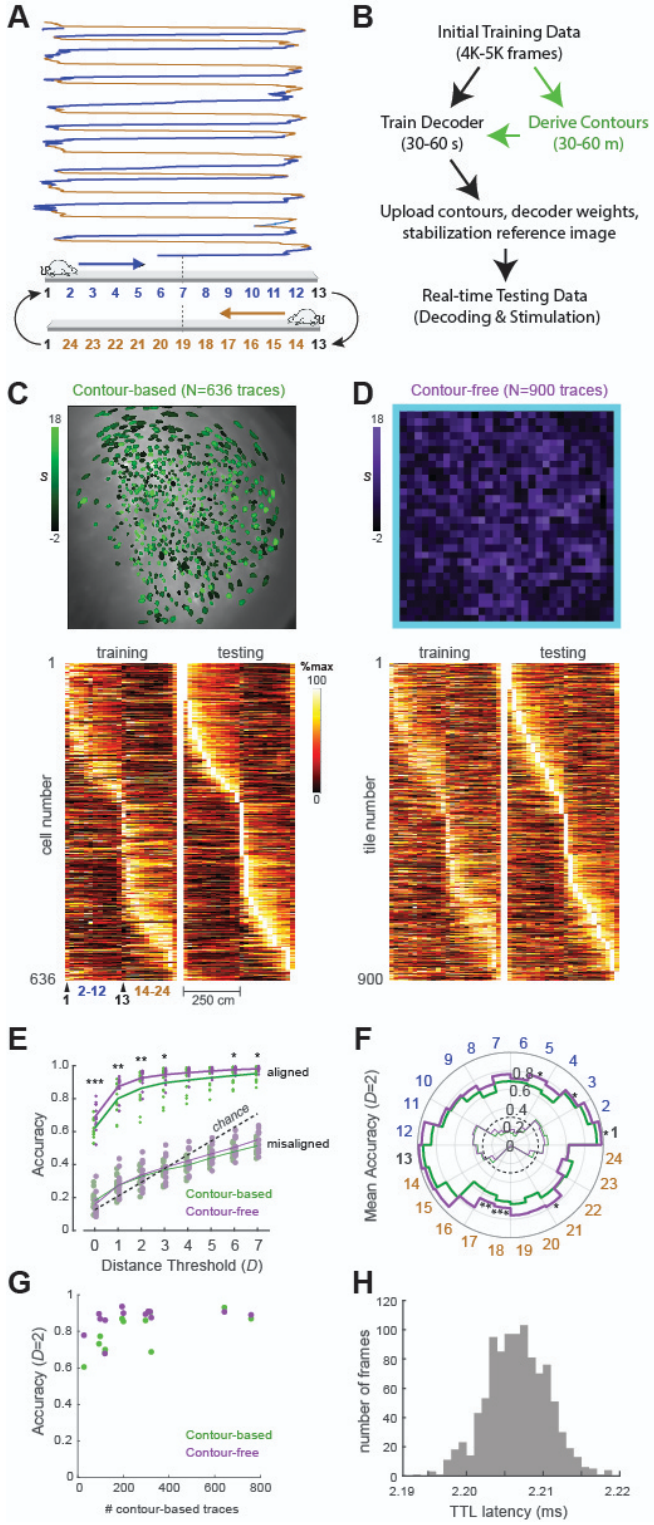


Figure 2. *Decoding position from hippocampal calcium activity*. *A*) The CA1 layer of the hippocampus was imaged while rats ran laps on a 250 cm linear track, subdivided into 24 position bins. *B*) Sequence of experimental steps for a real-time imaging & decoding session. *C*) *Top*: Cell contours identified within the image window by the CAIMAN algorithm in one rat; intensity of green shading indicates the similarity score,*S*, for each contour. *Bottom*: Heatmaps plot tuning curves (with no speed filtering; see Methods) of real-time calcium traces extracted from cell contours during the training (left) and testing (right) periods of the imaging session; both heatmaps are sorted on tuning peaks from the testing period. *D*) *Top*: Mosaic of square pixel mask tiles used for contour-free trace extraction in the same rat from 'C'; intensity of purple shading indicates *S* values for each tile, and edge tiles (light blue) were omitted from trace extraction. *Bottom*: Heatmaps of tuning curves for contour-free traces extracted from tiles. *E*) Decoding accuracy (y-axis) in all position bins as function of *D* (x-axis) for each rat (dots) and averaged over rats (lines); contour-free and contour-based classifers were both significantly more accurate ($p<.0001$ for all *D*) than chance or control classifiers trained on calcium traces misaligned with position data. *F*) Polar plot of mean decoding accuracy (radius) per position bin (angle) for *D*=2. *G*) Scatter plot shows accuracy of contour-free and contour-based traces plotted against the number of contour-based traces in each rat; for rats with low cell counts, contour-free decoding tended to outperform contour-based decoding. *H*) Distribution of latencies for decoding 1,024 online calcium traces. Asterisks in 'E' and 'F' show significance for the comparison between contour-based and contour-free decoding; *$p<.05$, **$p<.01$, ***$p<.001$.

pyramidal neurons behave as “place cells” that fire selectively when an animal traverses specific locations in space[18], so a rodent’s position can be reliably decoded from CA1 population activity[2,3,19]. Our testing used a virtual sensor (see Methods) that was capable of feeding stored image data to the ACTEV firmware at 22.8 fps, exactly as if raw video data were arriving from the MiniLFOV in real time. This allowed different firmware algorithms to be compared and benchmarked on the same stored datasets. Results obtained with the virtual sensor were verified to be identical with those obtained in real-time.

During a real-time session, an experimenter carries out four steps (Fig. 2B): 1) collect an initial imaging dataset and store it to the host PC, 2) pause for an intermission to identify cell contours (if necessary) and train a linear classifier using the initial dataset (see below), 3) upload classifier weights from the host PC to the Ultra96, and 4) perform real-time decoding with the trained classifier. To mimic these steps using the virtual sensor, image data from 13 rats (one session per rat) was collected and stored, yielding ~7 min (8K-9K frames) of sensor and position tracking data per rat. The linear classifier was trained on data from the first half of each session, and tested on data from the second half.

ACTEV extracted online fluorescence traces using one of two methods: *contour-based* or *contour-free*. Contour-based extraction derived calcium traces by summing fluorescence within cell contours identified during constrained non-negative matrix factorization (CNMF) by the CaImAn pipeline[20], which was applied to the training dataset during the intermission period (Fig. 2C; Supplementary Video 1). After contour identification, the training dataset was fed through a simulator to reconstruct online calcium traces that would have been extracted in real time during the first half of the session using the contour masks; simulated traces were then used to train the decoder. By contrast, contour-free extraction summed fluorescence within pixel masks obtained by partitioning the 512x512 image into a 32x32 sheet of tiles (excluding 124 tiles bordering the frame to prevent artifacts from motion stabilization), yielding 900 contour-free traces per rat (Fig. 2D; Supplementary Video 2).

After calcium traces had been extracted from the training dataset by either method, they were aligned with position tracking data and the classifier was trained to predict the rat's position from the traces. Training the classifier on 4K-5K frames of image data required <60 s of computing time on the host PC during the intermission period. However, contour-based trace extraction required an additional 30–60 min to identify cell contours and simulate online traces prior to training. Faster methods for contour identification may be possible, but contour-free trace extraction eliminated this delay at no cost to decoding accuracy (see below).

After the linear classifier had been trained, learned weights were uploaded from the host PC to the Ultra96; image data from the second half of the experimental session was then fed through the virtual sensor for real-time decoding. The classifier's output vector consisted of 23 units that used an ordinal scheme (see Methods) to represent 24 binned locations (12 per running direction) along the track (Fig. 2A, bottom). Decoding accuracy was quantified by measuring the percentage of frames for which the decoded position was within $\pm D$ bins of the rat's true position (Fig. 2E). Averaged over rats, mean decoding accuracy was higher for contour-free than contour-based traces (Figs. 2E,F), even though contour-based pixel masks were better aligned with single neurons. Analogous results have been reported in electrophysiology, where decoding is sometimes more accurate from "cluster-free" multiunit activity than from single-unit spikes[21].

One reason that contour-based decoding was less accurate than contour-free decoding was because in our dataset, the number of contour-based traces (which differed among rats) was always less than the number of contour-free traces (fixed at 900 per rat); the largest contour-free decoding advantages were seen in rats with <400 contour-based traces (Fig. 2G). CaImAn can detect more contours per rat if sensitivity parameters are adjusted, but CNMF derives cell contours by 'demixing' fluorescence signals from one another to pick out activity from single neurons; this process does not consider whether pixels contain information about behavior, so it may exclude pixels that contain decodable position information. Accordingly, we

found that decoding position from CaImAn's offline (demixed) traces did not improve accuracy over decoding from online traces derived from the same cell contours (Extended Data Fig. 2). Accurate decoding minimally requires that calcium traces must retain stable spatial tuning across the session, so spatial tuning curves were derived for calcium traces during the first (training) versus second (testing) half of the session. A similarity score (S) was computed for each trace using the formula $S = -\log_{10}(P) \times \text{sign}(R)$, where R is the correlation between the trace's tuning curves from training versus testing, and P is the significance level for R. Contour-based traces had higher mean S scores in every rat, so the average contour-based trace had better spatial tuning than the average contour-free trace (paired t_{12}=7.02, p=1.4e^{-5}); however, contour-free traces (including those with high S scores) were greater in number, allowing them to convey more information at the population level (Extended Data Fig. 3). Contour-free pixel masks with high S scores (bright purple squares in Fig. 2D) were sometimes located in regions of the image where CaImAn detected no cell contours at all (green pixel masks in Fig. 2C). Based on these results, the contour-free approach is recommendable as a more efficient online trace extraction method for accurate population decoding in real time.

After the decoder had been trained, real-time classifier output from each frame was relayed via ethernet to the host PC for storage and visualization (Supplementary Video 2), and was also used to trigger TTL feedback outputs from the Ultra96. The latency to generate feedback from decoded calcium traces was measured as the time elapsed between acquiring the last pixel of each cropped frame and the rising edge of the triggered TTL pulse. Under a worst-case scenario of decoding the maximum number of calcium traces supported by the hardware (N=1,024), mean decoding latency was 2,206±4 μs and never exceeded 2,220 μs (Fig. 2H). This highlights the advantages of our FPGA-based hardware design; it would be difficult to trigger feedback at such short intervals and low variability if online image data were processed on a CPU or GPU running programs with multiple threads, or if real-time video were

relayed to the image processor through USB or ethernet instead of by a direct hardware connection with the DAQ.

In summary, our system provides the research community with a first-of-its-kind hardware platform for online calcium imaging that can perform real-time decoding of neural population activity and generate closed-loop feedback at millisecond latencies. Source files for all hardware, software, and firmware are openly available at https://github.com/zhe-ch/ACTEV.

ACKNOWLEDGEMENTS

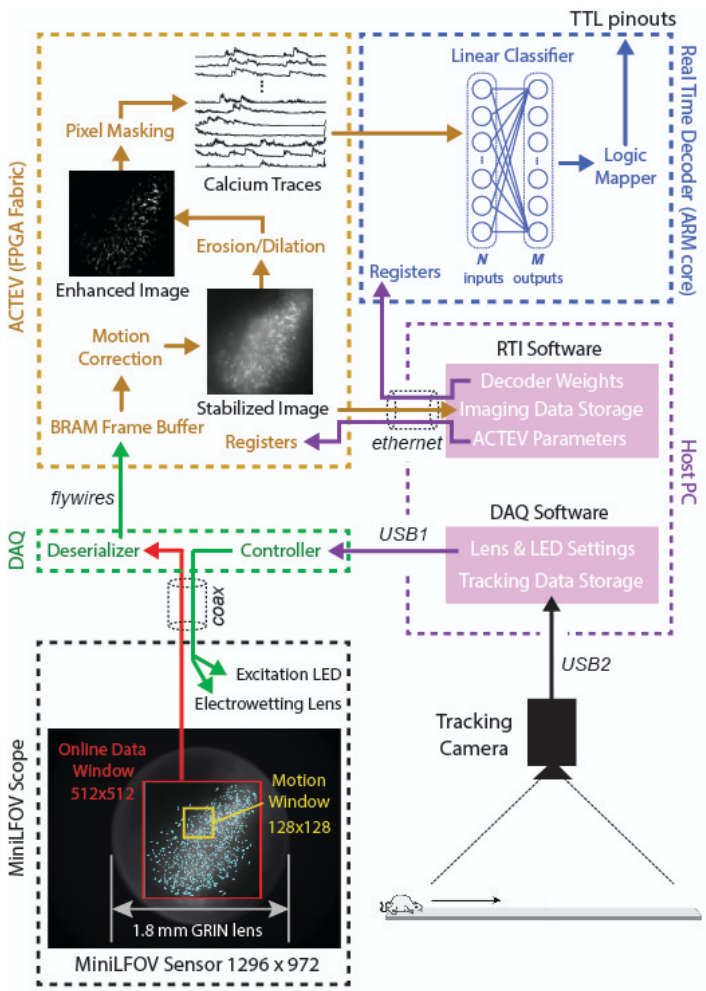
The authors thank Xilinx Corporation for donation of the Ultra96 board. We also thank Jeffrey Johnson for technical assistance in designing the custom interface board, and Shiyun Wang and Ryan Grgurich for helpful comments on the manuscript. This work was supported by NSF NeuroNex 17040708 (HTB, PG, DA, JC), 1UF1NS107668 (DA), and MH122800 (AI, HTB).

METHODS

Hardware and software

Our system is designed for use with UCLA Miniscope devices[3] (available at miniscope.org) and other miniscopes[15,16,17] that use the UCLA DAQ interface. The DAQ hardware requires modification by soldering flywires to the PCB (see Fig. 1); instructions for doing this are available at https://github.com/zhe-ch/ACTEV. In addition to the miniscope and DAQ hardware, our system also requires the Avnet Ultra96 board (available from the manufacturer at Avnet.com) mated to our custom interface board for receiving video from the Miniscope DAQ (which can be ordered from a PCB manufacturer with design files available at https://github.com/zhe-ch/ACTEV). Our ACTEV firmware (for online image stabilization, image enhancement, calcium trace extraction) and embedded software (for real-time decoding and

ethernet communication, which runs under the FreeRTOS operating system) must be programmed onto the Ultra96 board by copying a bootable bitstream file to a MicroSD card and then powering up the board with the card inserted. The MiniLFOV version of this bitstream file (used for experiments reported here) is available at https://github.com/zhe-ch/ACTEV. Bitstream files for other miniscope models, as well as Vivado™ source code for generating new bitstream files, are available on request. Real-time imaging and decoding is controlled by custom RTI software (available at https://github.com/zhe-ch/ACTEV) which runs on the host PC alongside standard DAQ software (available at miniscope.org).



Extended Data Fig. 1. *Flow chart of data processing and control pipelines for real time imaging*. Serialized video data from the MiniLFOV is transmitted through a coaxial tether cable to the DAQ, where it is deserialized and transmitted via the flywire bus to our ACTEV (Accelerator for Calcium Trace Extraction from Video) firmware programmed on the field-programmable gate array (FPGA) of the Ultra96. When used with the MiniLFOV (as in performance tests reported here), ACTEV crops the 1296x 972 sensor image down to a 512x512 subwindow (selected to contain the richest area of fluorescing neurons in each individual rat) before storing video frames to a BRAM buffer on the FPGA. Subsequent processing steps are described in the methods text.

Image stabilization

To correct for translational movement of brain tissue, a 128x128 pixel region with distinct anatomical features must be selected to serve as a motion stabilization window (this is done prior to initial data acquisition). ACTEV's image stabilization algorithm[8] rigidly corrects for translational movement of brain tissue by convolving the 128x128 stabilization window in each frame with a 17x17 contrast filter kernel, and then applying a fast 2D FFT/IFFT based algorithm to correlate the window contents with a stored reference template (derived at the beginning of the experimental session) to obtain a 2D motion displacement vector for the current frame. Supplementary Video 3 demonstrates performance of ACTEV's real-time motion stabilizer.

Background removal

After motion stabilization, ACTEV performs a sequence of three filtering steps[9] to enhance the 512x512 image: 1) smoothing via convolution with a 3x3 mean filtering kernel, 2) estimating the frame background via erosion and dilation with a 19x19 structuring element[22], and 3) subtracting the estimated background from the smoothed image. These filtering steps yield an enhanced image in which fluorescing neurons stand out in contrast against background (see Supplementary Video 1 and "Enhanced Image" in Extended Data Fig. 1).

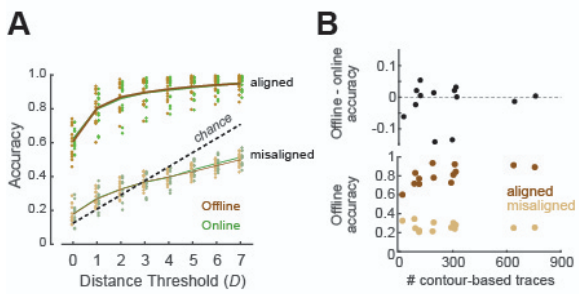
Calcium trace extraction

The enhanced image is filtered through a library of pixel masks to extract calcium traces from identified regions of interest (ROIs) in the 512x512 window. As explained in the main text, pixel masks can be created using either a *contour-based* or a *contour-free* approach.

Contour-based pixel masks

Pixel masks for contour-based trace extraction were derived by using the CaImAn[20] pipeline (implemented in python) during the intermission period to analyze motion-corrected sensor images from the training period (as noted in the main text, deriving contours in this way took 30-60 min of computing time on the host PC). CaImAn is an offline algorithm that uses constrained non-negative matrix factorization (CNMF) to isolate single neurons by demixing their calcium traces. CNMF is acausal so it cannot be used for real time trace extraction at short latencies, but during offline trace extraction, CaImAn generates a set of spatial contours surrounding pixel regions from which demixed traces were derived. Our system converts these contours into pixel masks for online extraction of contour-based traces. Once pixel masks had been identified from spatial contours, stored sensor data from the training period was passed through a simulator that used ACTEV's causal algorithm for extracting calcium traces from contour pixel masks. This yielded a set of calcium traces identical to those that would have been extracted from the training data in real time by ACTEV. These simulated traces were then used as input vectors to train the linear classifier.

Since the online traces are not generated by CNMF (and are thus not demixed from one another), they are susceptible to contamination from fluorescence originating outside of contour boundaries. However, position decoding from CaImAn's offline (demixed) traces was not more accurate than decoding from online (non-demixed) traces derived from the same contours (Extended Data Fig. 2), so crosstalk contamination had little effect upon decoding accuracy.



Extended Data Figure 2. *Decoding position from offline vs online contour-based traces*. *A*) Decoding accuracy (y-axis) in all position bins as function of *D* (x-axis) for each rat (dots) and averaged over rats (lines); classifers trained on aligned data were more accurate than those trained on misaligned data ($p<.0001$ for all *D*), but decoding accuracy did not differ for offline versus online traces derived from the same contours ($p>.45$ for all *D*). *B*) *Top*: Scatter plot shows offline minus online accuracy for contour-based traces (positive values on the y-axis indicate more accurate decoding from offline traces). *Bottom*: Offline contour-based decoding accuracy (y-axis) for a classifier trained on traces that were aligned or misaligned (control) with position data.

Contour-free pixel masks

To implement contour-free trace extraction, we simply partitioned the 512x512 image frame into a 32x32 sheet of tiles, each measuring 16x16 pixels on a side (Fig. 2D). No traces were extracted from 124 tiles bordering the edge of the frame, to avoid noise artifacts that might arise from edge effects in the motion stabilization algorithm. Hence, a total of 1,024-124=900 pixel mask tiles were used for contour-free calcium trace extraction. These traces were derived in real time and stored to the host PC throughout the initial data acquisition period, so they were immediately available for training the linear classifier at the start of the intermission. Hence, an advantage of contour-free trace extraction is that the intermission period between training and testing is shortened to just a few minutes, because the lengthy process of contour identification is no longer required. A disadvantage of contour-free trace extraction is that contour tiles do not align with individual neurons in the sensor image. As reported in the main text, this lack of alignment between neurons and pixel masks did not impair (and sometimes even enhanced) position decoding, but it places some limits upon what can be inferred from decoding results about how single neurons represent information about behavior.

Fluorescence summation

After pixel masks have been created using one of these two methods and uploaded to the FPGA, the calculations for extracting calcium traces from the masks are the same, regardless of what method was used to create the masks. Each mask specifies a set of pixels over which grayscale intensities are summed to obtain the fluorescence value of a single calcium trace:

$$T(f) = \sum_{i=1}^{P} p_i(f)$$

where $T(f)$ is the summed trace intensity for frame f, and $p_i(f)$ is the intensity of the ith pixel in the mask for frame f, and P is the number of pixels in the mask. Each contour-free mask was a square tile containing 16x16=256 pixels, whereas the size of each contour-based mask depended upon where CaImAn identified single neurons in the image. The mean size of a contour-based mask was 20-50 pixels, an order of magnitude smaller than contour-free tiles. In a few rats, a small amount of jitter occasionally penetrated the online motion correction filter, causing the stabilized sensor image to slip by 1-2 pixels against stationary contour masks during 1-2 frames (see Supplementary Video 3, line graph at lower right). This slippage misaligned pixels at the edges of the contour mask, producing intermittent noise in the calcium trace that was proportional to the fraction of misaligned pixels in the contour, which in turn was inversely proportional to the size of the contour, so that motion jitter caused more noise in traces derived from small contour masks than large contour masks.

Drop filtering

To filter out this occasional motion jitter noise from calcium traces, a *drop filter* was applied to traces derived from contour masks that contained fewer than 50 pixels. The drop filter exploited the fact that genetic calcium indicators have slow decay times, and therefore, sudden drops in trace fluorescence can be reliably attributed to jitter noise. For example, the GCamp7s indicator used here has a half decay time[18] of about 0.7 s, so at the MiniLFOV's 22.8 Hz frame rate, a fluorescence reduction of more than 5% between frames can only arise from jitter artifact. The drop filter defines a maximum permissible reduction in fluorescence between successive frames as follows:

$$\delta = GCq,$$

where *G* is the maximum possible fluorescence intensity for any single pixel (255 for 8-bit grayscale depth), *C* is the number of pixels in the contour mask for the trace, and *q* is a user-specified sensitivity threshold, which was set to 0.9 for all results presented here. The drop-filtered calcium trace value for frame *f* was given by:

$$F(f) = max[T(f), F(f - 1) - \delta].$$

It should be noted that while drop filtering protects against artifactual decreases in trace values, it offers no protection against artifactual increases that might masquerade as neural activity events. However, motion jitter almost never produced artifactual increases in fluorescence for the small contours to which drop filtering was applied, because small contours were "difficult targets" for patches of stray fluorescence to wander into during jitter events that rarely exceeded 1-2 pixels of slippage. Larger contours were slightly more likely to experience artifactual fluorescence increases during jitter events, but in such cases, the artifact was diluted

down to an inconsequential size because it only affected a tiny fraction of the large contour's pixels. In summary, jitter artifact was highly asymmetric, producing artifactual decreases but not increases for small contours, and producing negligible artifact of any kind for large contours.

Spike inference

Once trace values have been extracted (and drop filtered if necessary), ACTEV incorporates a real-time spike inference engine that can convert raw calcium trace values into inferred spike counts for each frame. However, for the CA1 data reported here, we found that the linear classifier was more accurate at decoding raw (unconverted) calcium traces than inferred spike counts (this was true for both contour-based and contour-free traces). This is not surprising, since the decay time constant for most genetically expressed calcium indicators (including the GCamp7s indicator used here[23]) is typically tens to hundreds of milliseconds; consequently, decoding information from raw calcium traces is tantamount to temporally integrating neural spike trains before decoding them. Temporal integration of spike trains is standard practice when decoding information from electrophysiological recordings of single-unit spikes (where it is common to derive neural firing rates by counting spikes in time bins ranging from tens to hundreds of milliseconds wide). Hence, by decoding information from raw calcium traces instead of inferred spike counts, we allow the GCamp molecule to perform temporal integration of neural activity, and thereby spare the computing hardware from having to consume resources on these integration calculations, which are standard for many neural decoding applications.

Storage of calcium traces and raw image data

Throughout the experimental session, raw image data (including real-time motion correction vectors) and calcium traces were transmitted via ethernet from the Ultra96 to the host

PC where they were received and stored by the RTI software. The Miniscope DAQ software ran concurrently with RTI software on the host PC, and was mainly used for adjusting the gain and focus of the Miniscope device. The DAQ software's default operation mode is to receive and display raw Miniscope video data from the DAQ via a USB 3.0 port (and store this data if the storage option has been selected), and also to receive and display raw behavior tracking video from a webcam through a separate USB port. At the start of a real experimental session, data acquisition by both programs (DAQ and RTI software) is initiated simultaneously with a single button click in the RTI user interface, so that Miniscope video storage by the RTI software and behavior video storage by the Miniscope DAQ software are synchronized to begin at exactly the same time. This allows behavioral data stored by the DAQ software to be aligned with Miniscope video and calcium trace data stored by the RTI software. During the intermission period between initial acquisition and real-time inference, data stored by the DAQ and RTI software is used to train the linear classifier on the host PC.

Training the linear classifier

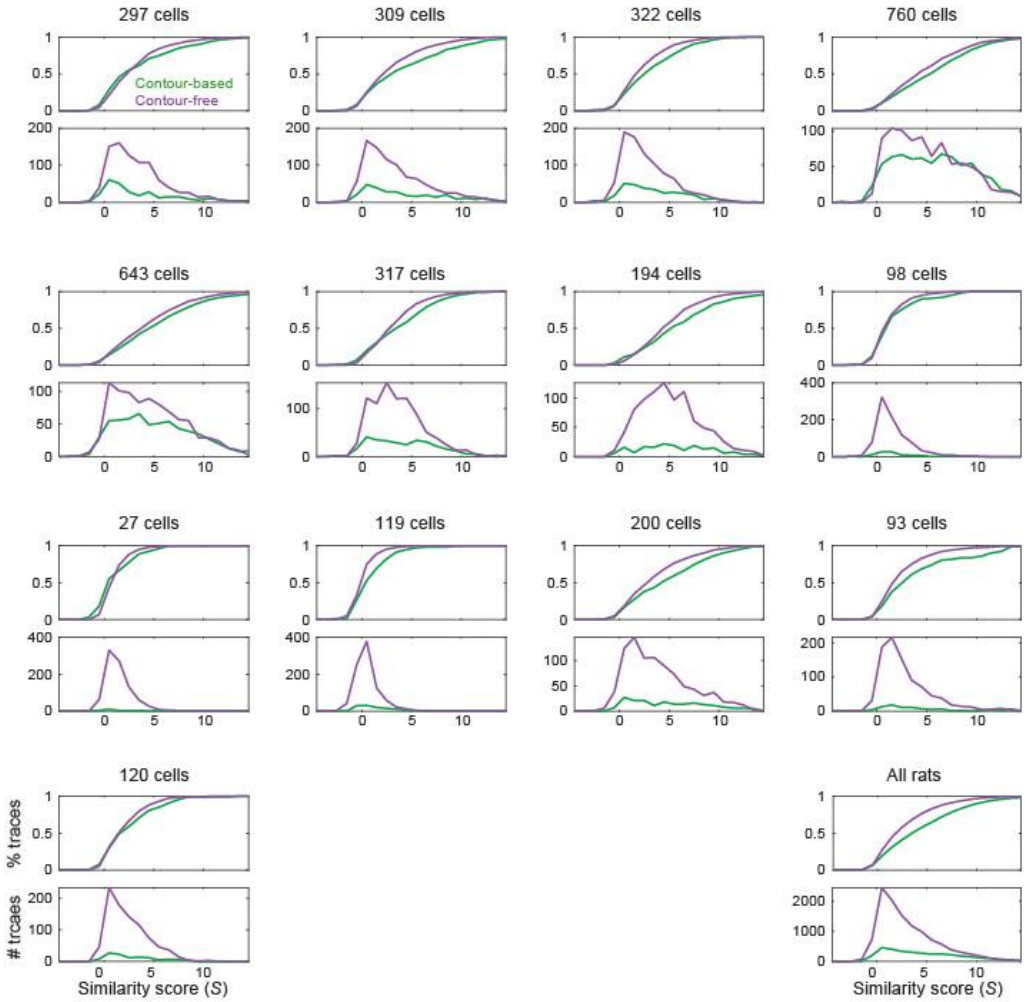
The linear classifier had N inputs and M outputs, where N is the number of calcium traces. The output layer utilized an ordinal scheme to represent each of the 24 binned track locations: at position bin #1 the target output for all output units was +1, at position bin #2 the target output was -1 for the first output unit and +1 for the remaining units, at position bin #3 the target output was -1 for the first two output units and +1 for the remaining units, and so on. Under this ordinal representation scheme, the total number of output units is equal to $M=K-1$, where K is the number of categories (position bins) to be encoded. Since the linear track was subdivided into $K=24$ spatial bins, there were $M=24-1=23$ binary classifier units in the decoder's output layer.

To train the linear classifier, data stored during the initial acquisition period by the DAQ and RTI software was analyzed during the intermission period by custom MATLAB scripts, available at https://github.com/zhe-ch/ACTEV. Training data consisted of calcium trace input vectors (derived by either contour-based or contour-free methods described above) and target position output vectors (encoding the rat's true position) for each Miniscope frame. These training vectors were passed to MATLAB's *fitceoc* function (from the machine learning toolbox) to compute the trained linear classifier weights, which were then uploaded from the host PC to the Ultra96 via ethernet, so that the real-time classifier could subsequently decode calcium traces in real time. As a control condition, calcium traces and tracking data were circularly shifted against one another by 500, 1000, 1500, 2000, or 2500 frames before training, and the trained classifier was then tested on correctly aligned trace inputs and target position outputs. Accuracy results from all five shift values were averaged together to obtain the 'misaligned' decoding accuracies plotted in Figs. 2E,F of the main text.

Decoding calcium traces

Once classifier weights had been uploaded to the Ultra96 from the host PC, it was then possible for the embedded ARM core in the Ultra96 MPSoC to read in calcium trace values from the DRAM in real time as each frame of data arrived from the Miniscope, and present these values as input to the linear classifier. The linear classifier performed calculations equivalent to MATLAB's *predict* function (from the machine learning toolbox) to generate an output vector predicting the rat's position on the track during each frame. This prediction was sent back to the host PC via ethernet for storage, and was also sent to a programmable *logic mapper* running on the ARM core (Fig. S1), which converted the linear classifer's output vector into a pattern of square pulses generated at TTL pinouts on the Ultra96. These TLL outputs are capable of driving closed-loop neural or sensory feedback via external devices such as lasers, electrical

stimulators, or audiovisual display equipment. For performance testing, we measured the latency to generate TTL outputs from decoded neural activity (see Fig. 1H in the main text), but we did not deliver closed loop feedback to our pilot rats.



Extended Data Fig. 3. *Spatial similarity score distributions by rat*. Each plot shows the cumulative probability density (top) and frequency distribution (bottom) of *S* scores from one rat; lower right plot shows *S* scores pooled over all rats. In every rat, the cumulative *S* distribution for contour-based traces is shifted to the right, indcating that contour-based traces were generally of higher quality (had more stable spatial tuning) than contour-free traces. But the number of contour-free traces (900 per rat) was always greater than the number of contour-based traces (shown at the top of each plot), and this was equally true for traces with high and low *S* values. Therefore, for position decoding, the lesser quality of individual contour-free traces was compensated by the fact that were greater in number, which was beneficial for population decoding.

Spatial tuning curves

To generate spatial tuning curves for fluorescence traces, each trace was first referenced to zero by subtracting it's minimum observed fluorescence value (over all frames) from the value in every frame: $Z(f) = T(f) - min(T)$, where $Z(f)$ is the zero-referenced trace value for frame f. The 250 cm linear track was subdivided into 24 evenly spaced spatial bins (12 each in the left-to-right and right-to-left directions; see Fig. 2A). Fluorescence activity, A, in each bin b was averaged by

$$A(b) = \sum_{i=1}^{B} Z(f_i)/B$$

where B is the number of visits to bin b and f_i is the ith frame during which the rat was visiting bin b. The tuning curve vector $[A_1, A_2, \ldots, A_{24}]$ was then normalized by dividing each of its elements by the value of the maximum element. Normalized tuning curve vectors were plotted as heatmaps in Figs. 2C and 2D. Spatial similarity scores (S) were computed as described in the main text; S distributions for each rat are shown in Extended Data Fig. 3.

When place cell tuning curves are derived offline from previously stored neural and position data, it is common practice to perform *speed filtering* that omits time periods when the rat is sitting still. This is done because during stillness, the hippocampus enters a "sharp wave" EEG state during which place cell activity is less reliably tuned for the animal's current location[24]. Here, such speed filtering was not implemented during online decoding of CA1 fluorescence, because the linear classifier's input layer received only real-time calcium trace data, and not position tracking data (which would be needed for speed filtering). To the extent that information about the animal's speed may have been encoded in some of the calcium traces, it is possible that some speed filtering may have been implicitly learned by the decoder

during training. But explicit speed filtering was not performed prior to training or testing the online classifier, and therefore was not performed prior to deriving tuning curves.

Imaging the hippocampal CA1 region in rats

Subjects

13 Long Evans rats (7F,6M) were acquired from Charles River at 3 months of age. Subjects were singly-housed within a temperature and humidity controlled vivarium on a 12 hour reverse light-cycle. Surgical procedures began after a one week acclimation period in the vivarium, and recordings and behavioral experiments began around 5 months of age. All experimental protocols were approved by the Chancellor's Animal Research Committee of the University of California, Los Angeles, in accordance with the US National Institutes of Health (NIH) guidelines.

Surgical procedures

Subjects were given two survival surgeries prior to behavior training in order to record fluorescent calcium activity from hippocampal CA1 cells. During the first surgery, rats were anesthetized with 5% isoflurane at 2.5 L/min of oxygen, then maintained at 2-2.5% isoflurane while a craniotomy was made above the dorsal hippocampus. Next, 1.2 μL of AAV9-Syn-GCamp7s (AddGene) was injected at .12μL/min just below the dorsal CA1 pyramidal layer (-3.6 AP, 2.5 ML, 2.6 DV) via a 10uL Nanofil syringe (World Precision Instruments) mounted in a Quintessential Stereotaxic Injector (Stoelting) controlled by a Motorized Lab Standard Stereotax (Harvard Apparatus). Left or right hemisphere was balanced across all animals. One week later, the rat was again induced under anesthesia and 4 skull screws were implanted to provide stable hold for the subsequent implant. The viral craniotomy was reopened to a diameter of 1.8 mm, and cortical tissue and corpus callosal fibers above the hippocampus

were aspirated away using a 27 and 30 gauge blunt needle. Following this aspiration, and assuring no bleeding persisted in the craniotomy, a stack of two 1.8 mm diameter Gradient Refractive INdex (GRIN) lenses (Edmund Optics) was implanted over the hippocampus and cemented in place with methacrylate bone cement (Simplex-P, Stryker Orthopaedics). The dorsal surface of the skull and bone screws were cemented with the GRIN lens to ensure stability of the implant, while the dorsal surface of the implanted lens was left exposed.

Two to three weeks later, rats were again placed under anesthesia in order to cement a 3D printed baseplate above the lens. First a second GRIN lens was optically glued (Norland Optical Adhesive 68, Edmund Optics) to the surface of the implanted lens and cured with UV light. The pitch of each GRIN lens was ≤0.25, so implanting two stacked lenses yielded a total pitch of ≤0.5. This half pitch provides translation of the image at the bottom surface of the lenses to the top while maintaining the focal point below the lens. This relay implant enables access to tissue deep below the skull surface. The MiniLFOV was placed securely in the baseplate and then mounted to the stereotax to visualize the calcium fluorescence and tissue. The baseplate was then cemented in place above the relay lenses at the proper focal plane and allowed to cure.

Linear alternation behavior

After rats had been baseplated, they were placed on food restriction to reach a goal weight of 85% *ad lib* weight and then began behavioral training. Time between the beginning of the surgical procedures and the start of behavior training was typically 6-8 weeks. Rats earned 20 mg chocolate pellets by alternating between two rewarded ends of a linear track (250 cm) during 15 min recordings beginning 5 days after baseplating. After receiving a reward at one end, the next reward had to be earned at the other end by crossing the center of the track.

Behavior tracking

A webcam mounted in the behavior room tracked a red LED located on the top of the Miniscope and this video was saved alongside the calcium imaging via the Miniscope DAQ software with synchronized frame timestamps. These behavior video files were initially processed by custom python code, where all the session videos were concatenated together into one tiff stack, downsampled to 15 frames per second, the median of the stack was subtracted from each image, and finally they were all rescaled to the original 8-bit range to yield the same maximum and minimum values before subtraction. Background subtracted behavior videos were then processed in MATLAB. The rat's position in each frame was determined using the location of the red LED on the camera. Extracted positions were then rescaled to remove the camera distortion and convert the pixel position to centimeters according to the maze size. Positional information was then interpolated to the timestamps of the calcium imaging video using a custom MATLAB script.

Histology

At the end of the experiment, rats were anesthetized with isoflurane, intraperitoneally injected with 1 mL of pentobarbital, then transcardially perfused with 100 mL of 0.01M PBS followed by 200 mL of 4% paraformaldehyde in 0.01M PBS to fix the brain tissue. Brains were sectioned at 40 μm thickness on a cryostat (Leica), mounted on gelatin prepared slides, then imaged on a confocal microscope (Zeiss) to confirm GFP expression and GRIN lens placement.

REFERENCES

1. Ghosh, K. K. *et al.* Miniaturized integration of a fluorescence microscope. *Nat. Methods* **8**, (2011).

2. Ziv, Y. *et al.* Long-term dynamics of CA1 hippocampal place codes. *Nat. Neurosci.* (2013) doi:10.1038/nn.3329.
3. Cai, D. J. *et al.* A shared neural ensemble links distinct contextual memories encoded close in time. *Nature* (2016) doi:10.1038/nature17955.
4. Aharoni, D., Khakh, B. S., Silva, A. J. & Golshani, P. All the light that we can see: a new era in miniaturized microscopy. *Nature Methods* (2019) doi:10.1038/s41592-018-0266-x.
5. Hart EE, Blair GJ, O'Dell TJ, Blair HT, Izquierdo A. Chemogenetic Modulation and Single-Photon Calcium Imaging in Anterior Cingulate Cortex Reveal a Mechanism for Effort-Based Decisions. (2020) J Neurosci. 40:5628-5643. doi: 10.1523/JNEUROSCI.2548-19.2020. PMID: 32527984; PMCID: PMC7363467.
6. Friedrich J, Zhou P, Paninski L. Fast online deconvolution of calcium imaging data. PLoS Comput Biol. 2017 Mar 14;13(3):e1005423. doi: 10.1371/journal.pcbi.1005423. PMID: 28291787; PMCID: PMC5370160.
7. Mitani A, Komiyama T. Real-Time Processing of Two-Photon Calcium Imaging Data Including Lateral Motion Artifact Correction. Front Neuroinform. 2018 Dec 18;12:98. doi: 10.3389/fninf.2018.00098. PMID: 30618703; PMCID: PMC6305597.
8. Chen Z, Blair HT, Cong J. (2019). LA-NorRMCorre: LSTM-Assisted Non-Rigid Motion Correction on FGPA for Calcium Image Stabilization. 27th ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA), 1-6 (02/24/2019).
9. Chen Z, GJ Blair, Blair HT, Cong J. (2020). BLINK: bit-sparse LSTM inference kernel enabling efficient calcium trace extraction for neurofeedback devices. Proceedings of the ACM/IEEE International Symposium on Low Power Electronics. ISBN:978-1-4503-7053-0
10. Chen Z, Zhou J, Blair GJ, Blair HT, Cong J. “Efficient kernels for real-time position decoding from in vivo calcium images.” IEEE International Symposium on Circuits and Systems (ISCAS), (2022).
11. Friedrich J, Giovannucci A, Pnevmatikakis EA. Online analysis of microendoscopic 1-photon calcium imaging data streams. PLoS Comput Biol. 2021 Jan 28;17(1):e1008565. doi: 10.1371/journal.pcbi.1008565. PMID: 33507937; PMCID: PMC7842953.
12. Aharoni, D. & Hoogland, T. M. Circuit investigations with open-source miniaturized microscopes: Past, present and future. *Frontiers in Cellular Neuroscience* (2019) doi:10.3389/fncel.2019.00141.
13. Grosenick L, Marshel JH, Deisseroth K. Closed-loop and activity-guided optogenetic control. Neuron. 2015 Apr 8;86(1):106-39. doi: 10.1016/j.neuron.2015.03.034. PMID: 25856490; PMCID: PMC4775736.
14. Zhang Z, Russell LE, Packer AM, Gauld OM, Häusser M. Closed-loop all-optical interrogation of neural circuits in vivo. Nat Methods. 2018 Dec;15(12):1037-1040. doi: 10.1038/s41592-018-0183-z. Epub 2018 Nov 12. PMID: 30420686; PMCID: PMC6513754.
15. Benjamin B. Scott, Stephan Y. Thiberge, Caiying Guo, D. Gowanlock R. Tervo, Carlos D. Brody, Alla Y. Karpova, David W. Tank (2018). Imaging Cortical Dynamics in GCaMP Transgenic Rats with a Head-Mounted Widefield Macroscope. Neuron, 100:1045-58.

16. de Groot, A. *et al.* (2020). Ninscope, a versatile miniscope for multi-region circuit investigations. *Elife* doi:10.7554/eLife.49987.
17. Guo C, Blair G, Sehgal M, Jimka FNS, Bellafard A, Silva AJ, Golshani P, Basso MA, Blair HT, Aharoni D. Miniscope-LFOV: A large field of view, single cell resolution, miniature microscope for wired and wire-free imaging of neural dynamics in freely behaving animals (submitted).
18. O'Keefe, J., and J. Dostrovsky. 1971. "The Hippocampus as a Spatial Map. Preliminary Evidence from Unit Activity in the Freely-Moving Rat." Brain Research 34 (1): 171–175. doi:10.1016/0006-8993(71)90358-1.
19. Wilson MA, McNaughton BL. Dynamics of the hippocampal ensemble code for space. Science. 1993 Aug 20;261(5124):1055-8. doi: 10.1126/science.8351520. Erratum in: Science 1994 Apr 1;264(5155):16. PMID: 8351520.
20. Giovannucci, Andrea, Johannes Friedrich, Pat Gunn, Jérémie Kalfon, Brandon L Brown, Sue Ann Koay, Jiannis Taxidis, et al. 2019. "CaImAn an Open Source Tool for Scalable Calcium Imaging Data Analysis." ELife 8: e38173. doi:10.7554/elife.38173.
21. Deng X, Liu DF, Kay K, Frank LM, Eden UT (2015). Clusterless Decoding of Position from Multiunit Activity Using a Marked Point Process Filter. *Neural Comput*. 27(7):1438-1460. doi:10.1162/NECO_a_00744
22. Lu J, Li C, Singh-Alvarado J, Zhou ZC, Fröhlich F, Mooney R, Wang F. MIN1PIPE: A Miniscope 1-Photon-Based Calcium Imaging Signal Extraction Pipeline. Cell Rep. 2018 Jun 19;23(12):3673-3684. doi: 10.1016/j.celrep.2018.05.062. PMID: 29925007; PMCID: PMC6084484.
23. Dana, H., Sun, Y., Mohar, B. *et al.* High-performance calcium sensors for imaging activity in neuronal populations and microcompartments. *Nat Methods* 16, 649–657 (2019). https://doi.org/10.1038/s41592-019-0435-6
24. Buzsáki, G. (2015). Hippocampal sharp wave-ripple: A cognitive biomarker for episodic memory and planning. *Hippocampus*, *25*(10), 1073–1188. https://doi.org/10.1002/hipo.22488

SUPPLEMENTARY VIDEO CAPTIONS

Supplementary Video 1. *RTI view of contour-based calcium trace extraction*. The online image display (left window) shows the motion corrected and enhanced (i.e., background subtracted) sensor image data as it arrives in real time from the Ultra96 board. The right window shows a scrolling display of 63 selected calcium traces from regions outlined by colored borders in the image display window. These traces are derived on the Ultra96 by summing fluorescence within

their respective contour regions, and the resulting trace values are transmitted (along with sensor image data) via ethernet to the host PC for display in the RTI window. For demonstration purposes, traces shown in the window on the right are normalized within the ranges of their own individual minimum and maximum values.

Supplementary Video 2. *Real-time decoding of contour-free calcium traces*. The online image display (left window) shows the motion corrected and enhanced mosaic of contour-free pixel mask tiles. The right window shows a scrolling heatmap display of 103 (out of the total 900) calcium traces with the highest tuning curve similarity scores, *S* (see main text). The calcium trace rows in the heatmap are sorted by the peak activity location of each trace, so that calcium activity can be seen to propagate through the population as the rat runs laps on the linear track. The line graph at the top shows the rat's true position (blue line) together with its predicted position (orange line) decoded in real time by the linear classifier. For demonstration purposes, trace intensities shown in the scrolling heatmap are normalized within the ranges of their own individual minimum and maximum values.

Supplementary Video 3. *Real-time motion correction*. The left and right windows show sensor video data before and after motion correction, respectively. The line graphs at bottom show x (yellow) and y (green) components of the image displacement between frames before (left) and after (right) motion correction.