A fast image processing toolbox for all-optical closed-loop control of neuronal activities

Weihao Sheng, Xueyang Zhao, Xinrui Huang, Yang Yang*

School of Life Science and Technology, ShanghaiTech University, Shanghai 201210, China.

* correspondence: yangyang2@shanghaitech.edu.cn

Abstract

The development of *in vivo* imaging and optogenetic tools makes it possible to control neural circuit activities in an all-optical, closed-loop manner [1], but such applications are limited by the lack of software for online analysis of neuronal imaging data. We developed ORCA (Online Real-time activity and offline Cross-session Analysis), which performs image registration, neuron segmentation, and activity extraction at over 100 frames per second, fast enough for real-time readout of neural activity. Using ORCA, we demonstrate closed-loop control of neurons in the mouse auditory cortex by two-photon targeted optogenetic inhibition of sound-evoked activity in selected neurons. ORCA also includes an offline module that processes hour-long imaging sessions within minutes and efficiently tracks neurons across multiple sessions. With a modular design, ORCA can be easily integrated into existing analysis toolboxes. In summary, ORCA is a powerful toolbox for closed-loop neuronal manipulation and fast data analysis.

Introduction

Establishing a causal relationship between neural activity and behavior is central to understand brain function. This endeavor has been greatly facilitated by optogenetics [2-8]. Currently, most optogenetic experiments use pre-determined stimulation protocols, without considering the ongoing activity of relevant neurons. However, as neuronal activities are highly heterogeneous and closely linked to brain states and behavior, fine-tuned perturbations based on real-time neuronal activities will better reveal the dynamic interaction between neural circuits and behavior [9-13]. Such closed-loop experiments, in which the input to the system (*i.e.*, optogenetic stimulation) depends on the output (*i.e.*, neural activity), are furthermore enabled by optical imaging techniques to monitor ongoing neural activities [14, 15]. However, it is still challenging

to identify active neurons and measure their activities from imaging data in real-time with high accuracy for closed-loop manipulations.

Processing microscopy data of neuronal activities typically involves several key steps. The first is image registration to compensate for the shift between successive image frames due to the relative motion between the brain and the microscope. The alignment quality affects the precision and reliability of subsequent processing. Although several methods are available, this time-consuming step needs substantial acceleration for online processing [16-18]. Another key step is online identification of active components. Current methods such as non-negative matrix factorization [19], graph cut algorithms [20], and convolutional neural networks [21] all require considerable computational power and imaging datasets spanning a relatively long time for effective identification of active components, thus not suitable for fast online processing of streaming images. The same limitations hold for other widely used imaging processing softwares [22-24]. In addition, these imaging analysis pipelines lack the feature to track automatically the same regions of interest (ROIs) across multiple imaging sessions. Imaging the same population of neurons *in vivo* for an extended period of time is now possible with genetically encoded indicators and the implantation of chronic imaging windows or GRIN lenses, for investigating the dynamics and plasticity of neural circuits [11, 25-27]. But manually identifying the same neurons from multiple imaging sessions is time-consuming and error-prone. In addition, some microscopes allow the user to rotate the optical axis. This feature adds flexibility to *in vivo* imaging, but at the same time exacerbates the difficulty of identifying the same neurons across imaging sessions captured with slightly different angles, as rigid transformation alone cannot correct such distortions [23, 28].

We developed an image processing toolbox, ORCA (Online Real-time activity and offline Cross-session Analysis), for online ROI identification and activity extraction from small datasets, and automatic offline cross-session ROI tracking for large, multi-session datasets. To validate its efficiency, we used ORCA to identify sound-responsive neurons in the mouse auditory cortex, and selectively suppressed the identified neurons with two-photon targeted optogenetic inhibition in a closed-loop setting. Each module of ORCA can be used independently, integrated to other systems, or together as a complete pipeline. The new features markedly accelerate imaging processing both in online and offline settings.

Results

Pipeline overview

ORCA consists of online and offline pipelines (Figure 1). The online pipeline detects active components from registered images, and plots the contours of identified ROIs as well as their activity traces within one second, providing real-time readouts of neuronal activities. Using similar algorithms, the offline pipeline extracts and plots neuronal activties from a complete imaging session containing thousands of frames within minutes, up to ten times faster than other methods. The offline pipeline also includes a cross-session alignment module for accurately tracking ROIs across multiple imaging sessions.

Fast and accurate GPU-based image registration

The first module, *Image Registration*, removes motion artifacts caused by brain pulsation and body movement. To achieve high processing speed, we implemented an algorithm to search for the optimal translation optimized for GPU (Graphics Processing Unit) acceleration (see Methods for details). We compared the processing speed of ORCA with single-step DFT [17], TurboReg [16], and moco [18], using an example movie of 5040 frames of 512×512 pixels, containing both high-frequency lateral shifts and gradual drifting in the X-Y plane (Figure 2a). ORCA outperforms the other methods, especially when running on GPU (Figure 2b), with comparable registration accuracy as shown by the magnified z-stack images (Figure 2c, top panels). Examination of the offset of each frame relative to the template image indicated that ORCA captures both small and fast shifts, and large and slow drifts (Figure 2c, bottom panels).

In order to quantitatively assess the registration accuracy of ORCA, a simulated movie was created by compiling a series of identical images with pseudo-random x and y shifts, termed “true shift values” (Figure 2d). Image registration on this movie was conducted using ORCA and the other three methods above. ORCA achieves similar or better performance, as quantified by the discrepancy between calculated shifts and the ground truth, suggesting that ORCA is a more efficient tool in image registration (Figure 2e & f).

Online identification and segmentation of active components

Most behavioral or neurophysiological experiments run in multiple repeats ("trials"; Figure 3a). A typical trial lasts a few seconds, as does the inter-trial interval (ITI). With a typical image acquisition rate of 10 ~ 30 frames/s, one trial generates a few dozen images. Adjusting optogenetic stimulation parameters based on neuronal activities in the preceding or the current trial requires identifying neurons and extracting their activities from a small set of images; the computation must be completed within seconds. To satisfy such requirements, we developed a novel variance scaling algorithm (see Methods for details).

To demonstrate ORCA's ability to identify active components online, we performed two-photon calcium imaging of neurons in the mouse auditory cortex (ACx). Each imaging trial lasted 4 s at 15 frames/s, generating 60 images; 1 s after imaging onset, a pure tone was played for 0.2 s (Figure 3a). The "baseline" period was defined as the 1 s before tone presenation, and the "response" period started with the tone onset and ended 1 s before the next trial begins. We first computed the cumulative activities of all pixels from 60 images (Figure 3b, "sum up activity"). Due to the small image number, the background fluorescence level fluctuates substantially; a simple thresholding algorithm will thus yield many false positives. To improve detection accuracy, we scaled each pixel's cumulative fluorescence level by the standard deviation of its fluorescence over time (Figure 3b, "multiply SD"). This simple algorithm dramatically increased the signal-to-noise ratio, enabling us to distinguish active components from the noisy background. To segment active components into ROIs, we performed thresholding based on user-defined ROI size and fluoresecent level (Figure 3b, "filtering"), and extracted the activity trace of each identified ROI (Figure 3b, "display"). Compared to previous methods using local correlation, our approach identified more active ROIs (Figure 3c). The whole identification and display process took less than 1 s (Figure 3d), well within the range of inter-trial intervals for behavioral experiments, and is fast enough for intra-trial manipulation. Activity traces of all identified ROIs can be calculated and plotted instantly (Figure 3e & f), and can also be exported to downstream control systems. We sorted the ROIs by their calcium activity in the descending order.

We noted that some labeled neurons were not identified by ORCA (Figure 3e). To validate the efficiency of ORCA to identify active neurons, we manually drew circles around visible neurons and inspected their calcium activities (Supp. Fig. 1). We found that none of the neurons unidentified by ORCA had significant activity (Supp. Fig. 2), which indicates that ORCA effectively captured all active neurons.

Offline segmentation and cross-session alignment for repeated imaging

ORCA provides two options for offline image analysis. For trial-based experiments, whole-session imaging data can be analyzed within minutes using the algorithm for online identification with one extra integration step: after active ROIs are extracted from each trial, they are integrated into a single mask for the whole session (Figure 4a). For non-trial-based experiments, a published algorithm, HNCcorr [20], is incorporated for offline ROI segmentation. HNCcorr can also be used for trial-based experiments by processing the whole imaging session. We compared the processing speed and identification accuracy of ORCA and HNCcorr in an example trial-based imaging session. In this experiment, ACx neurons were imaged while a series of different pure tones were played. From this imaging session, ORCA and HNCcorr identified 29 and 28 ROIs, respectively, with 20 overlapping ROIs (Figure 4c, Supp. Fig 3). ORCA only missed one tone-responsive neuron, which had extremely low baseline fluoresence (Figure 4c, white arrow; Supp. Fig. 4). The processing speed of ORCA is over 200 times faster than HNCcorr (Figure 4b, 50 seconds compared to 4 hours, for 2880 image frames).

One challenge in imaging analysis is to track the same neurons across multiple imaging sessions obtained over an extended period of time. Different sessions may have slightly different imaging angle due to imperfect adjustment of the microscope objective's orientation relative to the plane of the imaging window, which hampers the identification of the same neurons across sessions. Furthermore, manual tracking is time-consuming and prone to human inconsistency. To address these issues, ORCA uses affine transformation, a linear mapping method that deals with uni-directional distortion, to correct for the angle differences between sessions (Methods). An example of two overlay imaging sessions of the same field of view (FOV) is shown in Figure 4d (left panel). The same neurons from different sessions may overlap completely or partially in the overlay image if they are in the center of the FOV, or not overlap if located in the periphery (Figure 4d, center panel). Our algorithm successfully aligned all the cells (Figure 4d, right panel). Masks from individual sessions were merged into one unified mask with a 'capture-all' strategy: if an ROI was identified in multiple sessions with slightly different shapes, its shape in the unified mask is the union of the shapes across all sessions (Figure 4e, green arrow and box); ROIs identified only in one or a few sessions were also incorporated in the unified mask

(Figure 4e, blue arrow and box). After indexing all ROIs in all imaging sessions, the data analysis module extracts and plots the neural responses across different sessions. As an example, we plotted the individual and averaged changes in calcium signals ($\Delta F/F$) of one ROI in two imaging sessions responding to pure tones of different frequencies (Figure 4f).

A demo for all-optical closed-loop control of neuronal activities

To demonstrate ORCA's capacity for closed-loop photostimulation, we incorporated ORCA into a Thorlabs Bergamo II two-photon microscope with a spatial light modulator (SLM) for online manipulation of tone-responsive neurons in the mouse ACx (Figure 5a). We labeled ACx neurons with genetically encoded calcium indicator and optogenetic silencer by co-injecting AAV2/9-Syn-Cre, AAV2/9-hSyn-FLEX-GCaMP6s, and AAV2/9-hSyn-DIO-hGtACR1-mCherry viruses (Supp. Fig. 5). We delivered a variety of pure tones (0.5 s each) to the mouse, and imaged neuronal calcium responses starting 1 s prior to the tone onset for a total of 4 s per tone presentation. The ITI was 1 s, common for trial-based sensory and behavioral experiments, such as Go/No-Go and 2-alternative-forced-choice tasks. ORCA identified ROIs and plotted calcium responses online within 1 s, before the onset of the next trial (Figure 5b). As shown in the ORCA screenshot, ROI1 was responsive to the 4k Hz pure tone. The ROI1 contour was loaded into the stimulation software for targeted optogenetic inhibition via the SLM (Figure 5a). The identification was accurate as ROI1 was completely inactivated (Figure 5b & 5c). Thus, ORCA is well-adapted to commercial imaging systems for activity-based closed-loop control.

Discussion

Combining serveral novel algorithms, we developed ORCA, an imaging analysis toolbox that processes a small set of imaging data quickly for closed-loop neuronal stimulation, and performs cross-session image analysis for large datasets with high speed and accuracy. We validated our toolbox with calcium imaging experiments, but the software is in principle equally suitable for imaging studies using other fluorescent sensors, such as acetylcholine, dopamine and serotonin sensors [29-32].

To date, most optogenetic experiments are still open-loop, using stimulation parameters that are pre-determined rather than activity-guided. Only recently have

neuronal activities been considered in guiding optogenetic manipulations, in the form of large-scale imaging data [12] or real-time activities of pre-identified neurons [13]. Nonetheless, an imaging analysis pipeline that identifies active neurons from ongoing imaging experiments, essential for effective manipulation of highly dynamic and heterogeneous neuronal populations, is still lacking. Such a pipeline needs to complete raw image registration, active component identification and activity extraction within a few seconds. Our GPU-based registration algorithm registers 60 frames (512×512 pixels) in only 0.2 s. To our knowledge, only one method using an external OpenCV package written in C++ achieves comparable registration speed [24]. Combined with our novel active component detection algorithm, our online pipeline finishes the entire analysis process and plots the ROI contours and response curves within 1 s. The online detection algorithm captures the majority (>90%) of active neurons in the FOV, missing only sparse neurons with extremely low base-level fluorescence. These features make ORCA suitable for closed-loop stimulation experiments with parameters updated trial-by-trial. Future optimization of the detection algorithm will help to achieve higher identification accuracy, possibly by further scaling down the detection threshold for neurons with extremely low base-level fluoresence and high temporal SD according to its neighboring fluorescence intensity.

Many studies have used long-term *in vivo* imaging to analyze the dynamics of the same population of neurons in multiple sessions. In such studies, neurons are typically tracked manually, which is error-prone especially with imaging angle differences. ORCA is the first to use spatial features of the original FOV to align different sessions with possible imaging angle discrepancies. Angle difference up to 5° can be corrected. Notably, angle differences that are too large will result in data loss especially in the margin, which cannot be fixed by post-processing. With its fast processing speed, the affline alighment function of ORCA can also be used for quality check and calibration during image acquisition. Users can first take a small number of images, calculate the angle offset from previous sessions, and adjust the microscope accordingly to match previous recordings. As more and more microscopes feature flexible imaging angles, ORCA is a good complement and widely applicable to these systems.

Segmenting neurons from noisy *in vivo* imaging data is challenging. To achieve high accuracy, segmentation methods often have to sacrifice processing speed and rely on large quantities of imaging data. In all-optical closed-loop experiments, however, data analysis requires fast processing using only small datasets. ORCA bridges the gap,

and can be flexibly implemented into existing imaging systems to facilitate application of closed-loop manipulations owing to its modular desgin. ORCA is also highly automated and user-friendly, requiring minimal coding experience from its users. Future optimization and faster processing.

Materials and methods

Data availability

Code used in this paper is available at https://github.com/YangYangLab/ORCA.

Animals

C57BL/6 mice were purchased from Slac Laboratory Animals (Shanghai, China). Mice were housed and bred in a 12 h light-dark cycle (7 am - 7 pm light) in the animal facility of ShanghaiTech University. Both male and female mice were used for the experiments. All procedures were approved by the Animal Committee of ShanghaiTech University.

Virus injection

AAV2/9-Syn-Cre, AAV2/9-hSyn-Flpo-WPRE-pA, AAV2/9-hEFla-fDIO-GCaMP6s, AAV2/9-hSyn-FLEX-GCaMP6s, and AAV2/9-hSyn-DIO-hGtACR1-P2A-mCherry-WPRE-pA were purchased from Taitool Co., Shanghai, China. For virus injection, mice were anaesthetized with isofluorane (induction, 4%; maintenance, 1-2%) and positioned onto a stereotaxic frame (Reward Co., Shanghai, China). Body temperature was maintained at 37℃ using a heat pad. Viruses were injected using a glass micropipette with a tip diameter of 15-20 μm through a small skull opening (< 0.5 mm^2) with a micro-injector (Nanoject3, Drummond Scientific Co., Broomall, USA). Stereotaxic coordinates for auditory cortex (ACx): 2.46 mm posterior to the Bregma, 4.5 mm lateral from the midline, and 1.2 mm vertical from the cortical surface. For calcium imaging experiments, we mixed AAV2/9-hSyn-Flpo-WPRE-pA and AAV2/9-hEFla-fDIO-GCaMP6s, with the final titer of 1.3×10^{12} and 2.1×10^{12} viral particles per ml, respectively. For closed-loop stimulation experiments, we mixed AAV2/9-Syn-Cre, AAV2/9-hSyn-FLEX-GCaMP6f-WPRE-pA, and AAV2/9-hSyn-DIO-hGtACR1-P2A-mCherry-WPRE-pA, with the final titer of 6.4×10^{9}, 6.9×10^{12} and 1.4×10^{12} viral particles per ml, respectively. We injected 0.2 μl virus mixture into the auditory cortex for all experiments, and waited 3-4 weeks before two-photon imaging experiments.

Cranial window implantation

Mice were anaesthetized with isoflurane (induction, 4%; maintenance, 1-2%) and positioned onto a stereotaxic frame (Reward Co.). Body temperature was maintained at 37℃ using a heat pad. Lidocaine was administered subcutaneously. The muscle

covering the auditory cortex was carefully removed with a scalpel. A 2×2 mm^2 piece of skull over ACx was removed, exposing the dura. The cranial window was sealed with a custom-made double-layered cover glass. UV-cure glue and dental acrylic were used to cement the cover glass onto the skull. A custom-made stainless steel headplate with a screw hole was embedded into the dental acrylic for head-fixed two-photon imaging.

Two-photon calcium imaging

Mice were injected with pentobarbital sodium (20 mg/kg) and head-fixed using the implanted headplate. Image series were taken at 15Hz with a two-photon microscope (Bergamo II, Thorlabs, Newton, NJ, USA) equipped with a 25X/NA 1.05 objective (Olympus, Kyoto, Japan) and a Ti:sapphire laser (DeepSee, Spectra-Physics, Santa Clara, USA) tuned to 940 nm. A fixed-wavelength laser (1040nm, Spectra-Physics, USA) and spatial light modulator (Thorlabs) were used for two-photon targeted optogenetic silencing. Mice were imaged when a series of pure tones (4kHz, 8kHz, 12kHz) were played. Sound was controlled using a state machine (Sanworks Co., Rochester, USA) and open-source Bpod software (Sanworks Co.).

Hardware for imaging data analysis

We used a Dell workstation to perform all computations. The workstation is equipped with two Intel Xeon Gold 5122 CPU, 64GB DDR4 RAM, and an Nvidia Quadro P6000 GPU, running Linux Debian 10. ORCA uses GPU to accelerate most processing steps, while CPU versions are also provided in the package. We recommend the GPU version for best performance.

Preparation and evaluation of image registration

To compare the speed of different algorithms, we performed registration in their default environments: single-step DFT and ORCA were tested in MATLAB R2020a (MathWorks, Natick, USA), while TurboReg and moco in Fiji (ImageJ 1.48v, Java 1.6.0_24, 64bit). File read/write time was excluded for better evaluation of computational efficiency of each algorithm. Timing of TurboReg, single-step DFT and ORCA started after files were loaded into the memory, and stopped as soon as algorithms produced registered results. Timing of moco was evaluated using a screen recording software (Kazam Screencaster) and inspected manually.

To quantitatively compare different algorithms, we generated a simulated movie as follows. An example movie consisting of 1200 frames was repeatedly registered using

moco, TurboReg (accurate mode) and ORCA for 10 times each. This movie was then carefully examined by tracking major features in the images and measuring their movements frame-by-frame, to ensure it reached maximum possible stability. Then, periodical shifts caused by heartbeats were simulated by adding directional movements using the following equation:

$$s_{x,i} = \begin{cases} 0, & \sin(\frac{2\pi}{F} i) < 0 \\ M_x * \sin(\frac{2\pi}{F} i), & \sin(\frac{2\pi}{F} i) \geq 0 \end{cases}$$

where F indicated frames-per-second during acquisition, M_x the maximum shifts in x direction, and i the current frame count. This kept all positive values in s_x and set all negative offsets to zero. An additional random distortion was also introduced using pseudo-random numbers as offsets. Thus shifts in x direction, s_x, became

$$s_{x,i} = \begin{cases} \mathrm{rand}(-M_r,\ M_r), & \sin(\frac{2\pi}{F} i) < 0 \\ M_x * \sin\left(\frac{2\pi}{F} i\right) + \mathrm{rand}(-M_r,\ M_r), & \sin(\frac{2\pi}{F} i) \geq 0 \end{cases}$$

where $\mathrm{rand}(-M_r,\ M_r)$ generated a pseudo-random sequence ranging from $[-\ M_r,\ M_r]$.

True shift values were generated in both x and y direction in the same manner with varying parametres, and simulated movie was tested among all four registration methods.

Registration accuracy was quantified as

$$\mathrm{diff}(t, e) = \sqrt{\sum_{i=1}^{n} (t_i - e_i)^2}$$

where t was the true values, and e was the estimated shifts generated by each algorithm. The registration accuracy was calculated separately for x and y directions.

Image registration algorithms in ORCA

Inspired by the moco algorithm [18], we combined multiple computational ideas to produce a faster implementation. Let a be an image frame of h rows (height) and w columns (width), and $a_{i,j}$ denote the pixel of the i-th row, j-th column. Let t be the template image (usually the first frame in an image series). Images t and a are standardized by mean-centering before registration:

$$a \coloneqq \frac{a - \bar{a}}{\mathrm{std}(a)}$$

$$t \coloneqq \frac{t - \bar{t}}{\mathrm{std}(t)}$$

Let Ms be the user-defined maximum shift value. We set its default value to 1/5 of the frame size. Users are advised to start with the default setting and adjust the parameter from there, as appropriate choice of Ms is important for the best performance. An Ms too large (e.g., for a 512*512 pixel image, Ms = 170) may result in faulty matches, while an Ms too small (e.g., for a 512*512 image, Ms = 1) will cause the registration to fail, because the x-y shift between frames is likely to exceed 1 pixel.

We search for the optimal x, y offsets within the maximum shifts to minimize the difference between a and Tm in their overlapping region, defined as

$$D(x, y) \coloneqq \frac{1}{(h - |x|)(w - |y|)} \sum_{i,j} (t_{i,j} - a_{i+x,j+y})^2$$

Substituting the denomiator $(h - |x|)(w - |y|)$ with $A_{x,y}$, the area of overlap between the template and the target frame, we can rewrite the equation as:

$$A_{x,y} \cdot D_{x,y} = \sum_{i+x,j+y} t_{i+x,j+y}^2 + \sum_{i,j} a_{i,j}^2 - 2 \sum_{i,j} a_{i,j} \cdot t_{i+x,j+y}$$

in which $D_{x,y}$ is the sum of all pixels squared in the overlapping area of the template, and i, j are corresponding indices. Given that $x, y \in [-ms, ms] \cap \mathbb{Z}$, there are finite possibilities of these values, and they can be calculated for any given x, y in the range beforehand.

To efficiently calculate these values, we register template t onto current frame a, so that $\sum t^2$ only need to be calculated once throughout the whole registration process. $\sum a^2$ and $\sum a \cdot t$, are both related to the current frame a. Notice that, once we determined the maximum shift ms, only a central fraction of the original images are then registered to the template. To further accelerate the computation, we crop out the marginal areas of the original image, using only the central part to register against the full template. The cropped image, a', should be

$$a' \coloneqq a_{i',j'}$$

Where $i' \in [Ms, h - Ms]$, and $j' \in [Ms, w - Ms]$. Thus we rewrite the equation into:

$$A' \cdot D_{x,y} = \sum_{i+x,j+y} t_{i+x,j+y}^2 + \sum_{i,j} a'^2_{i,j} - 2 \sum_{i,j} a'_{i,j} \cdot t_{i+x,j+y}$$

In this equation, a' are the same cropped image for any valid $x, y \in [-ms, ms] \cap \mathbb{Z}$,

and the overlapping area A' becomes static. The only dynamic part in the equation thus becomes $\sum a \cdot t$. $\sum a \cdot t$ is the convolution of a and $\tilde{t}$ evaluated at (0,0). Letting $\tilde{t}$ be the rotation of t by 180 degrees, we have

$$\sum_{i,j} a_{i,j} \cdot t_{i+x,j+y} = \text{conv2}\ (a, \tilde{t})$$

Convolution can be efficiently calculated using two-dimensional fast Fourier transform (2D-FFT):

$$\text{conv2}\ (a, \tilde{t}) = \text{ifft2}\ (\text{fft2}\ (a) \cdot \text{fft2}\ (\tilde{t}))$$

and thus we developed a highly optimised `conv2` function (`conv2_fftvalid.m` in the source code) for our registration, which is much faster than MATLAB's default implementation using basic addition and multiplication.

$\sum a^2$ can also be calculated as

$$\sum_{i,j} a_{i,j}^2 = \text{conv2}\ (a^2, \text{ones}(h - 2Ms, w - 2Ms))$$

where $\text{ones}(h - 2Ms, w - 2Ms))$ is a matrix of size $(h - 2Ms, w - 2Ms)$ with all elements being 1, in the MATLAB notation.

The whole registration process can be further accelerated by downsampling and doing a local search after upsampling. Running on GPU, our code is much faster than most other registration algorithms.

Online trial-by-trial identification of active components

We combined several statistical tools to produce fast segmentation of active ROIs on a trial-by-trial basis, each trial consisting of dozens of frames that constitute a multi-frame image. The image is first normalized by subtracting mean value and divided by the standard deviation of the whole image:

$$x := \frac{x - \bar{x}}{\text{std}(x)}$$

where x represents any data point in the image. Then, cumulative activity of all pixels are calculated by summing up all values in their temporal trace to form a two dimensional time-average image. Values below zero are substituted by zeros.

$$c_i := \begin{cases} 0, & \Sigma < 0 \\ \sum_t x_{i,t}, & \Sigma \geq 0 \end{cases}$$

Here, c_i represents an individual pixel in the 2-D time-average image, where i indicates the location of that pixel.

To increase the signal-to-noise ratio, we use standard deviation of each pixel's temporal trace as a scaling factor, and multiply by c_i. The product matrix undergoes a simple filter that converts it to a binary image using a user-defined threshold (default value set at 0.05), and eliminates areas smaller than a user-defined size (default set at 16 pixels). Pixels containing 1("positive pixels") indicates a masked region, and ROIs are generated by combining neighbouring positive pixels.

Offline segmentation of cell-like components using HNCcorr

We incorporated the MATLAB version of HNCcorr [20], an algorithm that automatically identifies active cell-like components, for non-trial-based imaging data. We wrote a simple wrapper function for interaction with HNCcorr.

Offline manual segmentation of components in a movie

Since automated identification may falsely identify unwanted components or neglect true active components, we provide a user-friendly interface for manual labelling of active components in a given movie, incorporated in the `mask_segmentation` module. Our manual labelling program allows users to play the movie for full length or just one trial, and add or erase components.

Cross session alignment

For multi-session imaging data of one FOV, we use time-average images to represent the imaging sessions, and define the time-average image of the first imaging session as the reference image. In a three-dimensional space, we define the xy plane as the plane of the reference image, and z axis perpendicular to xy. We then define the target image to be the time-average of a different imaging session. For sessions without rotation on the z axis, displacements in x and y axis were corrected simply by moving target images in x, y direction for optimum alignment. For sessions with rotation, because the image plane preserves the collinearity of the FOV, displacements can be corrected using affine transformation. We first computed similarity between the reference image and the target image, then applied different transformations onto the target image and search for maximum similarity between them. We used Matlab function `imregtform()` with '`rigid`' or '`affine`' parameters, and applied the transformation matrix by the function. We incorporated manual inspection and refinement by the users in this module. After user confirmation of the alignment results, masks with segmented ROIs from different sessions were combine to generate a

superset of ROIs. These ROIs were then transformed to the original session based on respective transformation matrices. ROIs in the margins may be cropped out in the cross-session alignment..

Activity extraction

As ORCA extracts neuronal activity trial-by-trial, the user needs to supply information about the trial structure, in particular timing parameters. The baseline activity F_0 of each ROI is the average of all pixels over a user-defined time window before stimulus onset:

$$F_0 := \frac{\sum_{i,t} a_i(t)}{i \cdot t}$$

where a_i is a pixel in the corresponding ROI, and $t \in [\text{Baseline Frames}]$. Then, the normalized activity of all ROIs is calculated by subtracting and dividing baseline activities:

$$F := \frac{\sum_i a_i}{n}, \quad F_{activity}(\mathrm{t}) := \frac{F(t) - F_0}{F_0}$$

Acknowledgements

We thank Drs. J. Lu, M. Ho, S.T. Han and M. Tsakiris for critical reading and editing of the manuscript. We thank the imaging facility of School of Life Science and Technology, ShanghaiTech University, for technical support. We also thank the following funding agencies (2018YFC1005004 and NSFC31970960 to YY) for the support of the work.

Competing interests

The authors declare no competing interests.

References

1. Grosenick, L., J.H. Marshel, and K. Deisseroth, *Closed-loop and activity-guided optogenetic control.* Neuron, 2015. **86**(1): p. 106-39.
2. Boyden, E.S., et al., *Millisecond-timescale, genetically targeted optical control of neural activity.* Nat Neurosci, 2005. **8**(9): p. 1263-8.
3. Boyden, E.S., et al., *Millisecond-timescale, genetically targeted optical control of neural activity.* Nature Neuroscience, 2005. **8**(9): p. 1263-1268.
4. Deisseroth, K., et al., *Next-generation optical technologies for illuminating genetically targeted brain circuits.* Journal of Neuroscience, 2006. **26**(41): p. 10380-10386.
5. Zhang, F., et al., *Optogenetic interrogation of neural circuits: technology for probing mammalian brain structures.* Nat Protoc, 2010. **5**(3): p. 439-56.
6. Packer, A.M., et al., *Two-photon optogenetics of dendritic spines and neural circuits.* Nature Methods, 2012. **9**(12): p. 1202-U103.
7. Yang, W.J. and R. Yuste, *Holographic imaging and photostimulation of neural activity.* Current Opinion in Neurobiology, 2018. **50**: p. 211-221.
8. Lutz, C., et al., *Holographic photolysis of caged neurotransmitters.*

Nature Methods, 2008. **5**(9): p. 821-827.

9. O'Connor, D.H., et al., *Neural coding during active somatosensation revealed using illusory touch.* Nat Neurosci, 2013. **16**(7): p. 958-65.

10. Prsa, M., G.L. Galinanes, and D. Huber, *Rapid Integration of Artificial Sensory Feedback during Operant Conditioning of Motor Cortex Neurons.* Neuron, 2017. **93**(4): p. 929-939 e6.

11. Peters, A.J., et al., *Reorganization of corticospinal output during motor learning.* Nature Neuroscience, 2017. **20**(8): p. 1133-+.

12. Robinson, N.T.M., et al., *Targeted Activation of Hippocampal Place Cells Drives Memory-Guided Spatial Behavior.* Cell, 2020. **183**(7): p. 2041-2042.

13. Zhang, Z., et al., *Closed-loop all-optical interrogation of neural circuits in vivo.* Nat Methods, 2018. **15**(12): p. 1037-1040.

14. Tian, L., et al., *Imaging neural activity in worms, flies and mice with improved GCaMP calcium indicators.* Nature Methods, 2009. **6**(12): p. 875-U113.

15. Chen, T.W., et al., *Ultrasensitive fluorescent proteins for imaging neuronal activity.* Nature, 2013. **499**(7458): p. 295-+.

16. Thevenaz, P., U.E. Ruttimann, and M. Unser, *A pyramid approach to subpixel registration based on intensity.* Ieee Transactions on Image Processing, 1998. **7**(1): p. 27-41.

17. Guizar-Sicairos, M., S.T. Thurman, and J.R. Fienup, *Efficient subpixel image registration algorithms.* Optics Letters, 2008. **33**(2): p. 156-158.

18. Dubbs, A., J. Guevara, and R. Yuste, *moco: Fast Motion Correction for Calcium Imaging.* Front Neuroinform, 2016. **10**: p. 6.

19. Pnevmatikakis, E.A., et al., *Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data.* Neuron, 2016. **89**(2): p. 285-99.

20. Speen, Q., et al., *HNCcorr: A Novel Combinatorial Approach for Cell Identification in Calcium-Imaging Movies.* Eneuro, 2019. **6**(2).

21. Soltanian-Zadeh, S., et al., *Fast and robust active neuron segmentation in two-photon calcium imaging using spatiotemporal deep learning.*

Proceedings of the National Academy of Sciences of the United States of America, 2019. **116**(17): p. 8554-8563.

22. Pachitariu, M., et al., *Suite2p: beyond 10,000 neurons with standard two-photon microscopy.* 2017: p. 061507.

23. Giovannucci, A., et al., *CaImAn an open source tool for scalable calcium imaging data analysis.* Elife, 2019. **8**.

24. Mitani, A. and T. Komiyama, *Real-Time Processing of Two-Photon Calcium Imaging Data Including Lateral Motion Artifact Correction.* Frontiers in Neuroinformatics, 2018. **12**.

25. Driscoll, L.N., et al., *Dynamic Reorganization of Neuronal Activity Patterns in Parietal Cortex.* Cell, 2017. **170**(5): p. 986-+.

26. Xin, Y., et al., *Sensory-to-Category Transformation via Dynamic Reorganization of Ensemble Structures in Mouse Auditory Cortex.* Neuron, 2019. **103**(5): p. 909-+.

27. Holtmaat, A., et al., *Long-term, high-resolution imaging in the mouse neocortex through a chronic cranial window.* Nature Protocols, 2009. **4**(8): p. 1128-1144.

28. Sheintuch, L., et al., *Tracking the Same Neurons across Multiple Days in Ca2+ Imaging Data.* Cell Reports, 2017. **21**(4): p. 1102-1115.

29. Feng, J., et al., *A Genetically Encoded Fluorescent Sensor for Rapid and Specific In Vivo Detection of Norepinephrine.* Neuron, 2019. **102**(4): p. 745-+.

30. Sun, F.M., et al., *A Genetically Encoded Fluorescent Sensor Enables Rapid and Specific Detection of Dopamine in Flies, Fish, and Mice.* Cell, 2018. **174**(2): p. 481-+.

31. Wan, J., et al., *A genetically encoded sensor for measuring serotonin dynamics.* Nat Neurosci, 2021.

32. Patriarchi, T., et al., *Imaging neuromodulators with high spatiotemporal resolution using genetically encoded indicators.* Nat Protoc, 2019. **14**(12): p. 3471-3505.

Figure 1

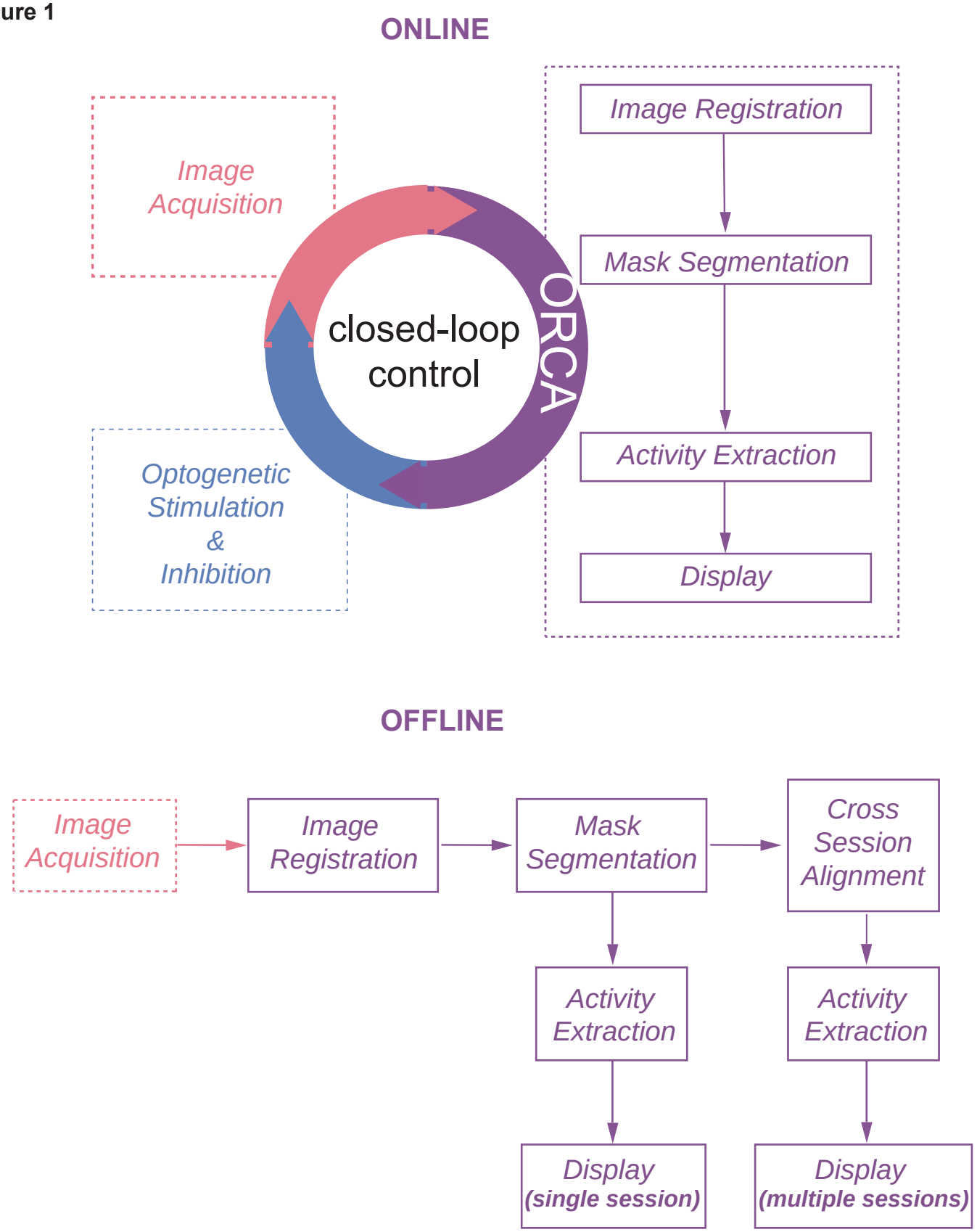


Figure 1: The online and offline pipelines of ORCA. The online pipeline can be incorporated into closed-loop control systems.

Figure 2

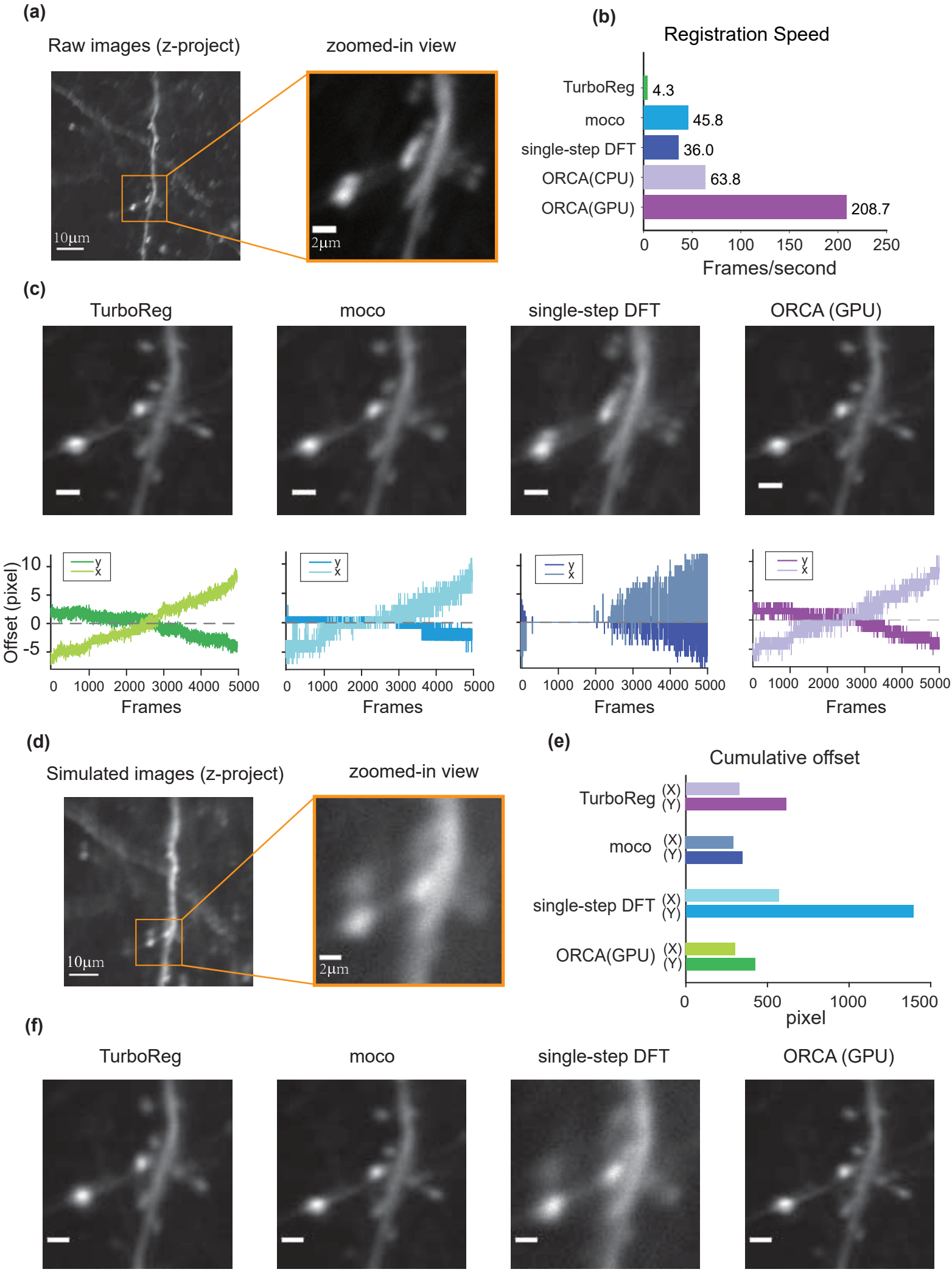

(a)
Raw images (z-project)
zoomed-in view
10μm
2μm
(b)
Registration Speed
TurboReg
moco
single-step DFT
ORCA(CPU)
ORCA(GPU)
4.3
45.8
36.0
63.8
208.7
0
50
100
150
200
250
Frames/second
(c)
TurboReg
moco
single-step DFT
ORCA (GPU)
Offset (pixel)
10
5
0
-5
y
x
0
1000
2000
3000
4000
5000
Frames
(d)
Simulated images (z-project)
zoomed-in view
10μm
2μm
(e)
Cumulative offset
TurboReg
moco
single-step DFT
ORCA(GPU)
(X)
(Y)
0
500
1000
1500
pixel
(f)
TurboReg
moco
single-step DFT
ORCA (GPU)

Figure 2: Raw image registration by ORCA and other softwares.
(a) Z-project of raw images and zoom-in view.
(b) Registration speed comparison for different methods.
(c) Top panel: z-project of zoomed-in view after registration; bottom: calculated shifts in x & y direction. Scale bar, 2um.
(d) Z-project of simulated movie.
(e) Quantification of registration accuracy with simulated data in (d).
(f) Z-project of zoomed-in view after registration of simulated movie. Scale bar, 2um.

Figure 3

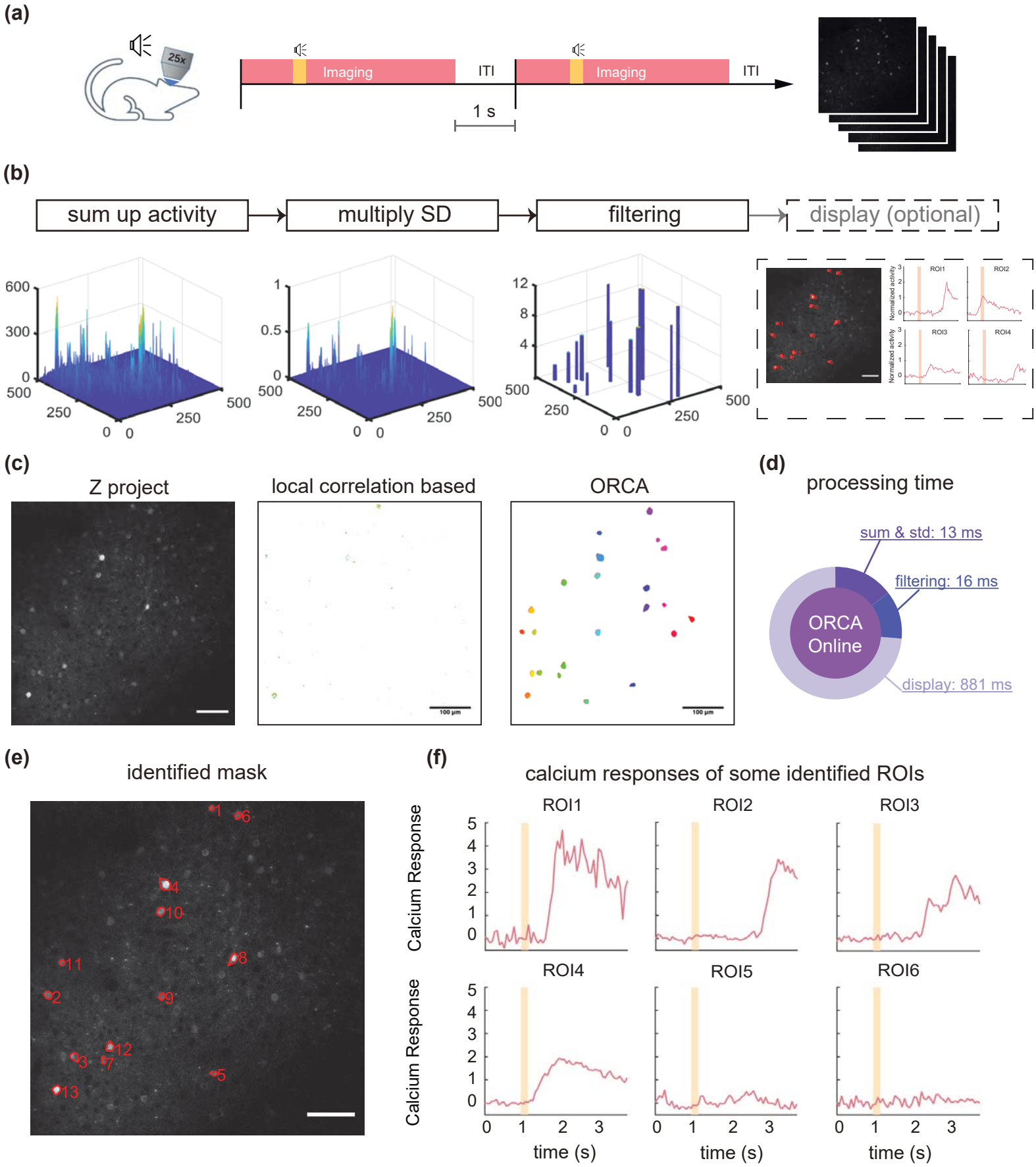


Figure 3: Online active neuron identification and extraction by ORCA.
(a) The flow of a typical trial-based imaging experiment.
(b) Online pipeline consists of 4 steps, with the final display step optional.
(c) ROIs identified by local correlation based algorithm (middle) and ORCA (right). Scale bar, 10um.
(d) Processing time of each step.
(e) An example of identified ROIs in one trial. scale bar, 50um.
(f) Calcium responses of ROIs identified in (e), sorted by calcium activity level. Orange shading, pure tone.

Figure 4

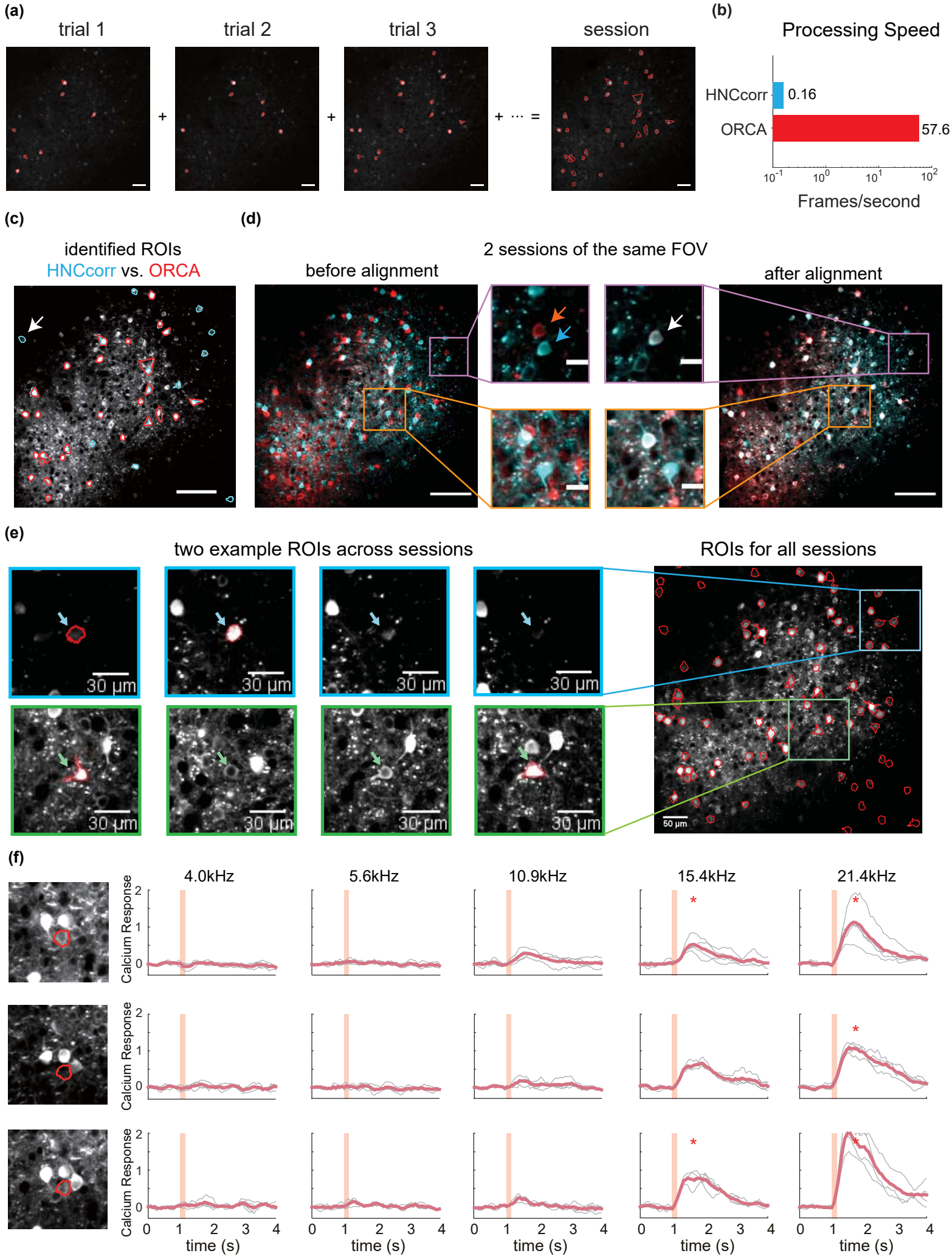

(a)
trial 1
trial 2
trial 3
session
+
+
+ ··· =
(b)
Processing Speed
HNCcorr
0.16
ORCA
57.6
10^{-1}
10^{0}
10^{1}
10^{2}
Frames/second
(c)
identified ROIs
HNCcorr vs. ORCA
(d)
2 sessions of the same FOV
before alignment
after alignment
(e)
two example ROIs across sessions
ROIs for all sessions
30 μm
50 μm
(f)
4.0kHz
5.6kHz
10.9kHz
15.4kHz
21.4kHz
Calcium Response
time (s)

Figure 4: Single- and multiple-session data analysis by ORCA.
(a) For single-session analysis, ROIs were identified in each trial, and summed up to generate the session mask.
(b) Processing speed of HNCcorr and ORCA for 512*512 images.
(c) ROI segmentation results of the same session by HNCcorr (blue) vs. ORCA (red). White arrow: the only active ROI identified by HNCcorr but missed by ORCA.
(d) Cross session alignment for two sessions of the same FOV. Red and cyan colours are z-project images of the two sessions. Zoomed-in view on the left shows displacements at both centre and margins, and on the right properly aligned ROIs by ORCA offline.
(e) Identified ROIs in continuous sessions.
(f) Calcium responses to different pure tones of one example ROI across three sessions. Orange shading indicates pure tone duration.

Figure 5

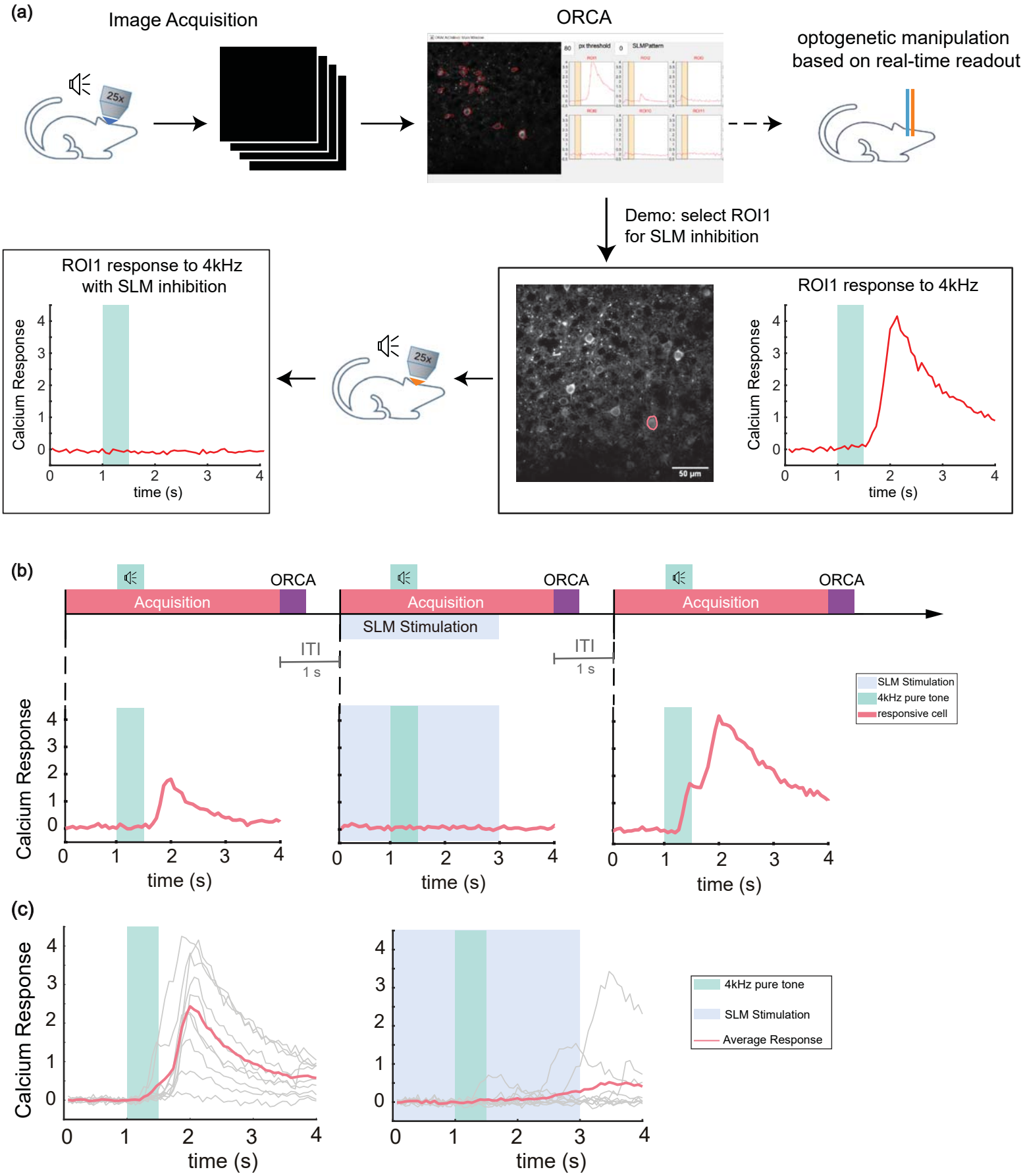
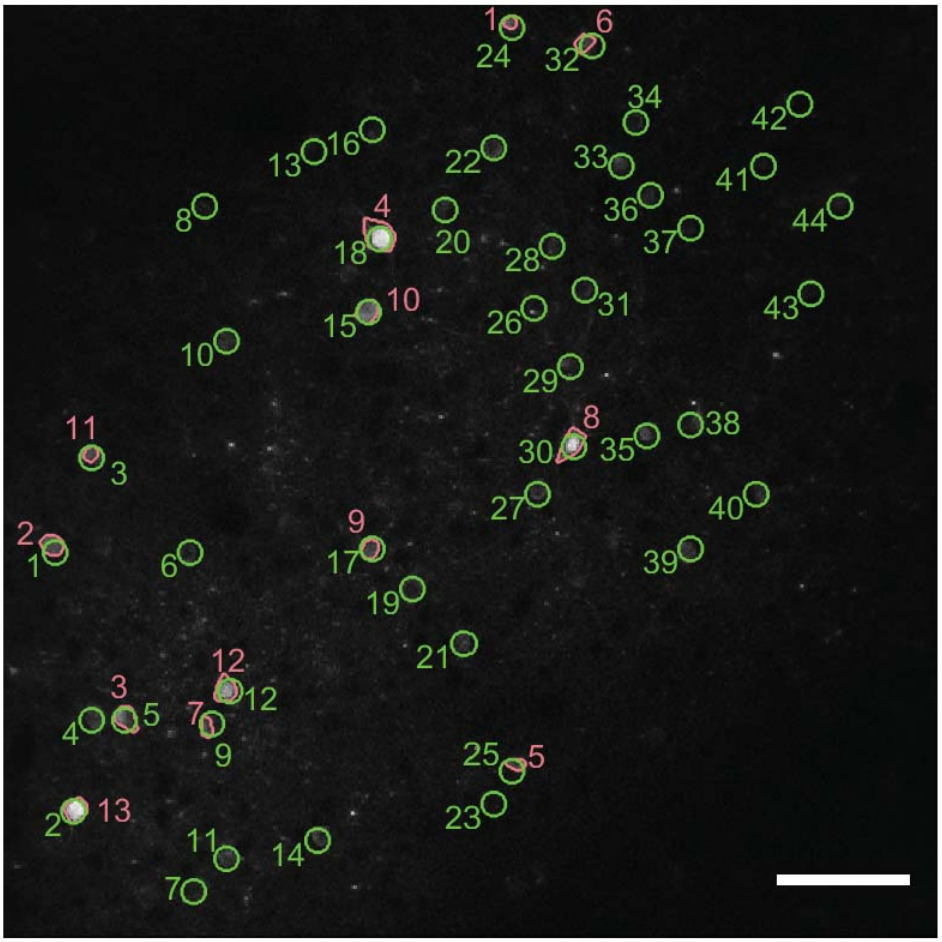


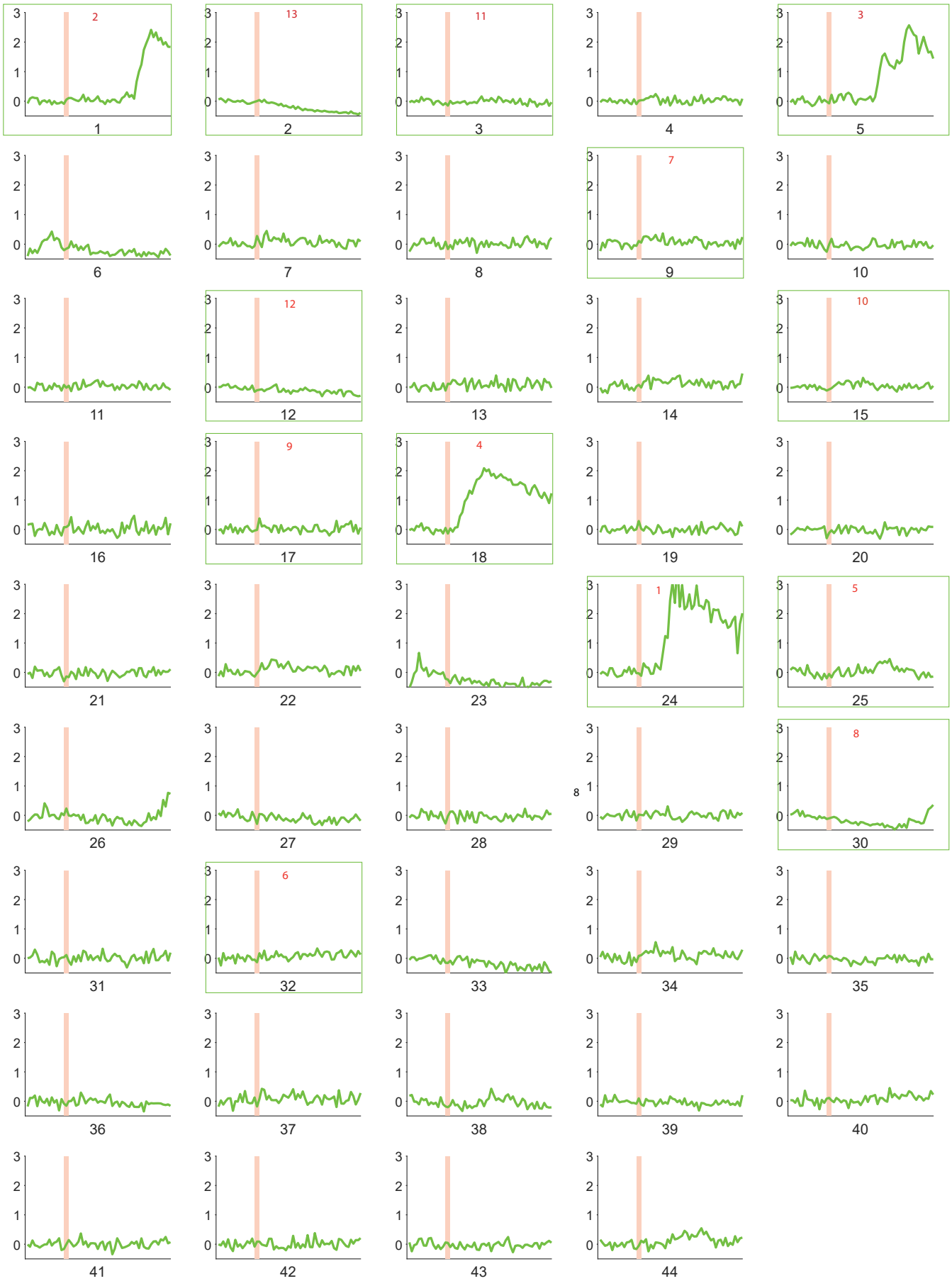
Figure 5: Application of ORCA in activity-based closed-loop experiments.
(a) Timeline of the experiment. ORCA identifies active ROIs, and feeds the selected identified ROI into the stimulation system for photoinhibition. This pipeline runs automatically without human inteference.
(b) Response of one example 4kHz-responsive cell to 4kHz pure tone with and without SLM inhibition.
(c) All responses of the cell in (b) Calcium responses to 4kHz with (right) or without (left) SLM inhibition.

Supp. Figure 1



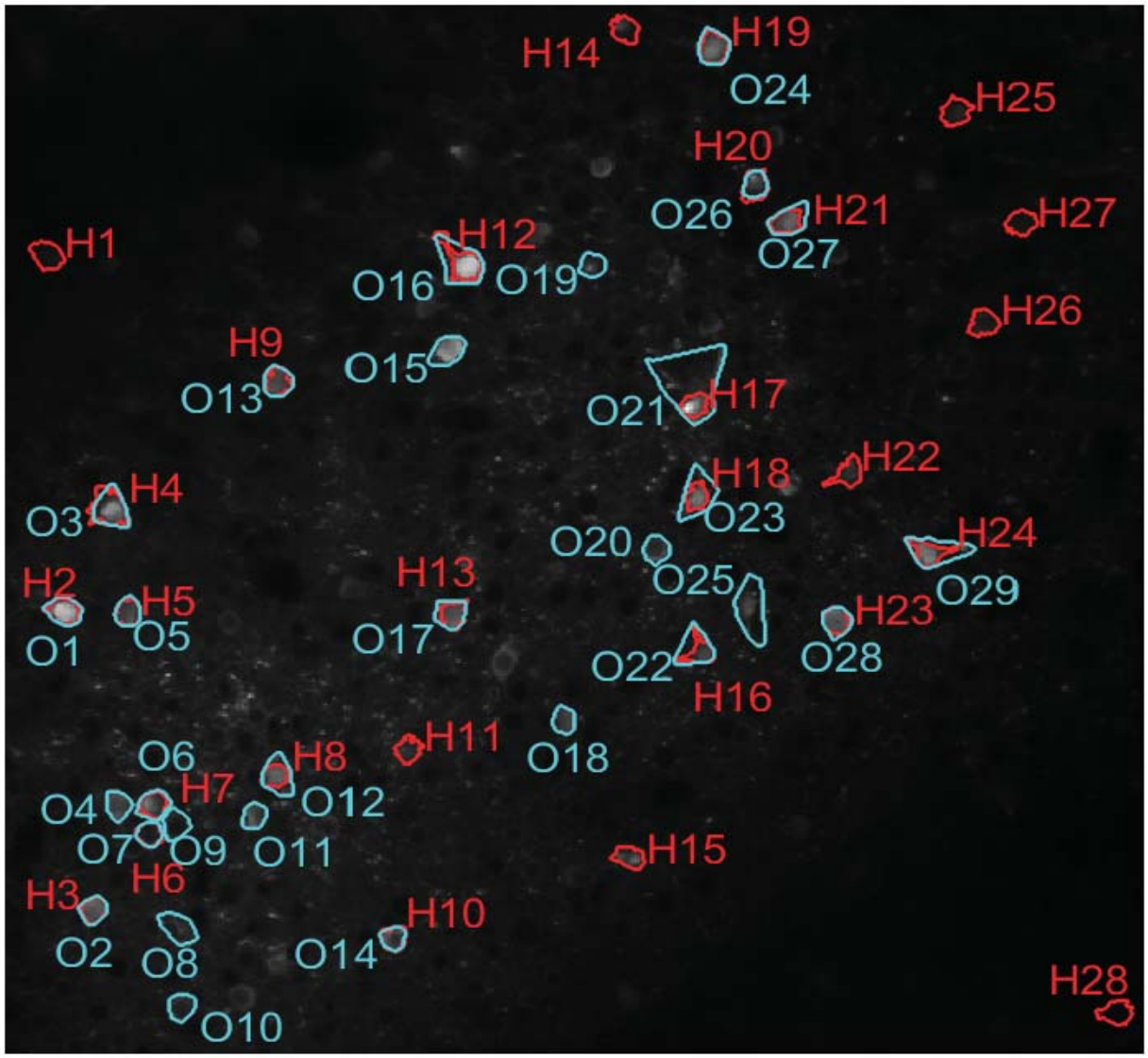
Supp. Figure 1: ROIs automatically detection by ORCA (red contours and numbers, same trial as in Figure 3), and manually drawn (green circles).

Supp. Figure 2



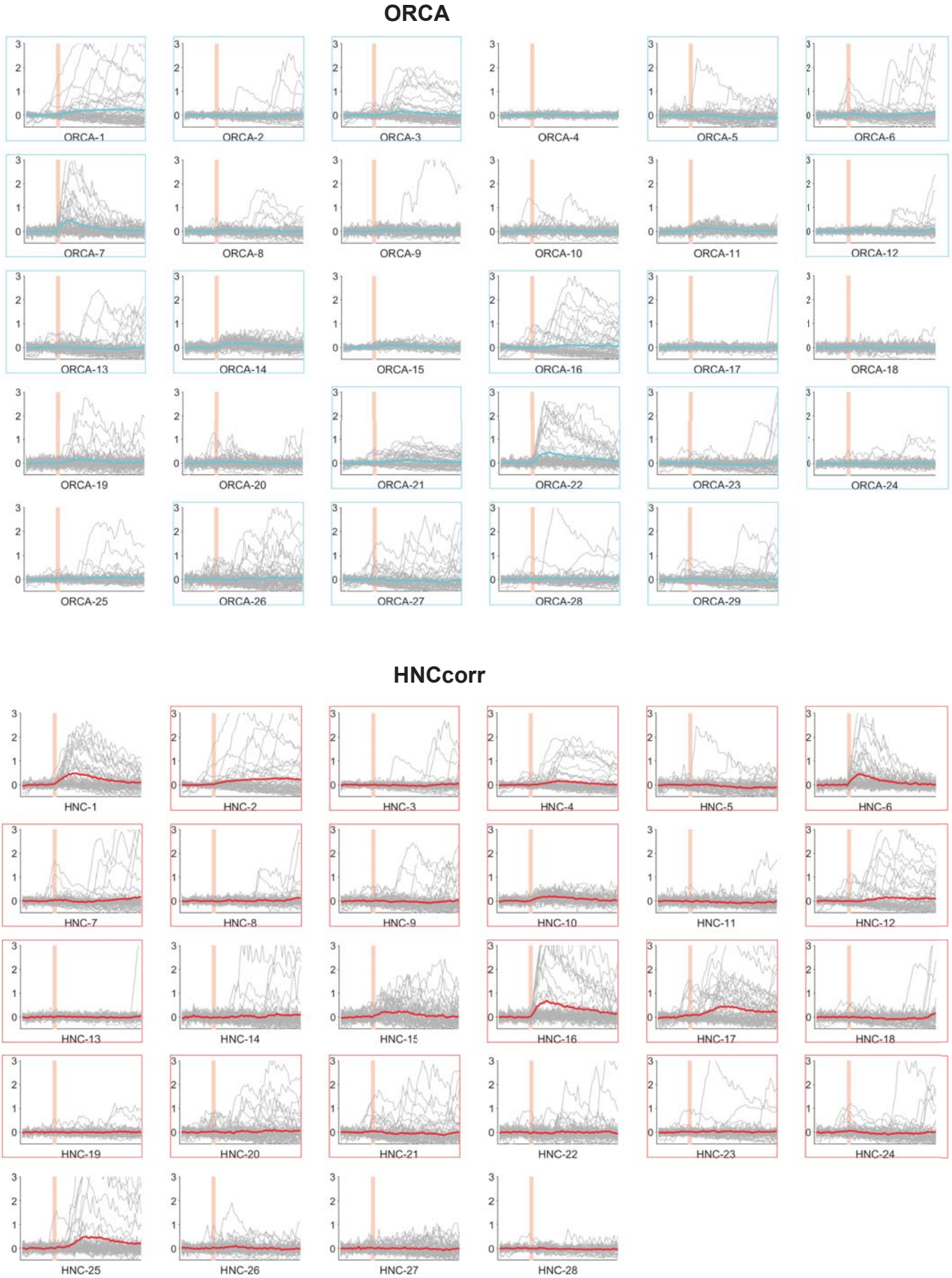
Supp. Figure 2: Calcium responses of each hand-drawn ROI shown in Supp. Fig. 1. The boxes indicated ROIs automatically identified by ORCA, indexed with red numbers. Orange shading indicates auditory stimulus (a pure tone).

Supp. Figure 3



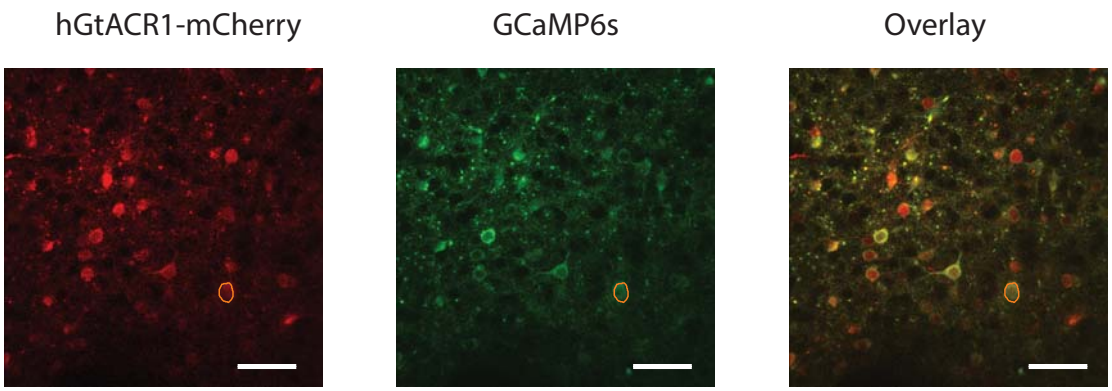
Supp. Figure 3: ROI segmentation results of one imaging session (same session as in Figure 4) by HNCcorr (blue) vs. ORCA (red). HNCcorr identified 28 ROIs, ORCA 29 ROIs, with 20 overlaps.

Supp. Figure 4



Supp. Figure 4: Calcium responses to pure tones of each ROI shown in Supp. Fig. 1. The boxes indicated overlapped ROIs. Orange shading indicates pure tone duration.

Supp. Figure 5



Supp. Figure 5: Expression of hGtACR1-mCherry (red) and GCaMP6s (green) in the FOV shown in Figure 5a. The circled neuron is the same neuron shown in Figure 5, which expresses both hGtACR1 and GCaMP6s. scale bar, 50 um.