



Fast and Accurate Motion Correction for Two-Photon Ca^{2+} Imaging in Behaving Mice

Weiye Liu^{1†}, Junxia Pan^{1†}, Yuanxu Xu^{1†}, Meng Wang², Hongbo Jia^{3,4}, Kuan Zhang^{1*}, Xiaowei Chen^{1*}, Xingyi Li^{2*} and Xiang Liao^{2*}

¹ Brain Research Center and State Key Laboratory of Trauma, Burns, and Combined Injury, Third Military Medical University, Chongqing, China, ² Center for Neurointelligence, School of Medicine, Chongqing University, Chongqing, China, ³ Advanced Institute for Brain and Intelligence, Guangxi University, Nanning, China, ⁴ Brain Research Instrument Innovation Center, Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences, Suzhou, China

OPEN ACCESS

Edited by:

Valentin Nägerl,
UMR 5297 Institut Interdisciplinaire
de Neurosciences, France

Reviewed by:

Vassily Tsytarev,
University of Maryland, College Park,
United States
Vijay Mohan K. Nambodiri,
University of California,
San Francisco, United States

*Correspondence:

Kuan Zhang
zhangkuan@tmmu.edu.cn
Xiaowei Chen
xiaowei_chen@tmmu.edu.cn
Xingyi Li
xingyi_li@cqu.edu.cn
Xiang Liao
xiang.liao@cqu.edu.cn

[†]These authors have contributed
equally to this work

Received: 09 January 2022

Accepted: 07 March 2022

Published: 26 April 2022

Citation:

Liu W, Pan J, Xu Y, Wang M,
Jia H, Zhang K, Chen X, Li X and
Liao X (2022) Fast and Accurate
Motion Correction for Two-Photon
 Ca^{2+} Imaging in Behaving Mice.
Front. Neuroinform. 16:851188.
doi: 10.3389/fninf.2022.851188

Two-photon Ca^{2+} imaging is a widely used technique for investigating brain functions across multiple spatial scales. However, the recording of neuronal activities is affected by movement of the brain during tasks in which the animal is behaving normally. Although post-hoc image registration is the commonly used approach, the recent developments of online neuroscience experiments require real-time image processing with efficient motion correction performance, posing new challenges in neuroinformatics. We propose a fast and accurate image density feature-based motion correction method to address the problem of imaging animal during behaviors. This method is implemented by first robustly estimating and clustering the density features from two-photon images. Then, it takes advantage of the temporal correlation in imaging data to update features of consecutive imaging frames with efficient calculations. Thus, motion artifacts can be quickly and accurately corrected by matching the features and obtaining the transformation parameters for the raw images. Based on this efficient motion correction strategy, our algorithm yields promising computational efficiency on imaging datasets with scales ranging from dendritic spines to neuronal populations. Furthermore, we show that the proposed motion correction method outperforms other methods by evaluating not only computational speed but also the quality of the correction performance. Specifically, we provide a powerful tool to perform motion correction for two-photon Ca^{2+} imaging data, which may facilitate online imaging experiments in the future.

Keywords: two-photon Ca^{2+} imaging, motion correction, behaving mice, image density feature, image registration, online experiment

INTRODUCTION

Two-photon microscopy is widely used for investigating neural functions at diverse scales, from the size of a single spine to a neuronal population (Dombeck et al., 2007; Grienberger and Konnerth, 2012; Huber et al., 2012). Recent advances in optical imaging techniques have enabled the monitoring of neural activity with high temporal resolution. However, movement of the brain, which is generally induced by behaviors such as licking and limb movements, remains a

serious limitation when recording animals that are awake (Greenberg and Kerr, 2009). Therefore, correction for brain movement enables the accurate analysis of morphology and activity for individual neurons in imaging data (Chen et al., 2013). In addition, when performing real-time processing during online functional imaging and photostimulation experiments (Griffiths et al., 2020), data processing must be performed with a high speed and robust performance. Therefore, a fast and accurate processing method for removing the movement artifacts from imaging data is a key requisite for online experiments (Mitani and Komiyama, 2018; Pnevmatikakis, 2019).

To reduce the image distortions caused by brain movements, frame-by-frame motion correction approaches have been developed based on image registration. An important set of image registration methods is based on image intensity. These algorithms are principally intuitive and provide good performance, but they are computationally expensive. As a representative intensity-based method, TurboReg (Thevenaz et al., 1998) has been integrated in ImageJ software and is widely used for correcting lateral motion artifacts. Moreover, to implement non-rigid transformation for the motion correction of Ca²⁺ imaging data, NoRMCorre (Pnevmatikakis and Giovannucci, 2017) was developed by splitting the imaging field into overlapping patches and estimating the transformation for each pixel *via* upsampling. The computational speed is slow for these motion correction methods, thus some efforts for improving the speed have been made. For instance, Moco (Dubbs et al., 2016) was programmed in Java and achieved faster motion correction than TurboReg by implementing a discrete Fourier transform and cache-aware upsampling. Similarly, fast Fourier transform was applied for processing motion correction in the Suite2p toolbox (Pachitariu et al., 2017).

An alternative type of image registration approach is based on feature detection. These approaches perform by extracting features through a global intensity gradient such as DoG (difference of Gaussians). In this way, the feature detection approaches to extracting features refer to representative structures. Thus, feature detection-based algorithms, such as SIFT (Lowe, 2004), ORB (Rublee et al., 2011), and AKAZE (Alcantarilla et al., 2013), have been used for image registration. However, these approaches are also relatively computationally expensive and perform unsatisfactorily in image registration applications.

Since the rise of deep learning methods, which have achieved state-of-the-art results in many research fields, deep learning-based image registration algorithms have been rapidly developed and used in many image registration applications (Haskins et al., 2020). However, these methods have also encountered the issue of high computational power demand (Thompson et al., 2020), which may limit their applications in online imaging experiments.

The current image registration-based motion correction methods are either computationally expensive or difficult to implement with a conventional computer, and some suffer from both disadvantages. In this work, we focus on rigid motion correction. As suggested in previous research (Mitani and Komiyama, 2018), rigid transformation is quite efficient when

imaging with resonant scanners; the high sampling rate can only induce negligible within-frame motion distortion. With the goal of improving both computational speed and accuracy, we propose a fast image feature extraction and registration (FIFER) method for motion correction of two-photon Ca²⁺ imaging data. Our proposed method is based on an image density estimation and clustering approach (Hinneburg and Keim, 1998), which is fast and robust in the extraction of imaging data features. The efficacy of the proposed motion correction method was first assessed with raw two-photon Ca²⁺ imaging data at both the neuronal population and single-spine scales. The quantitative evaluation of motion-corrected imaging data demonstrates that our method provides superior performance in comparison to other motion correction approaches, and thus provides a powerful solution to facilitate online functional imaging and photostimulation experiments in neuroscience research.

MATERIALS AND METHODS

Data Acquisition

Adult (8–12 weeks old) male C57BL/6J mice were obtained from the Laboratory Animal Center at the Third Military Medical University. All experimental procedures related to the use of animals were approved by the Third Military Medical University Animal Care and Use Committee and were conducted in accordance with institutional animal welfare guidelines.

For two-photon imaging in head-fixed awake mice (Li et al., 2018), we first glued a titanium head post to the skull for head fixation under isoflurane anesthesia. Three days after surgery, animals received 1 ml of water supply per day for 2–3 days and then underwent training and testing sessions with water deprivation in their home cages. For imaging experiments with Cal-520 AM (Tada et al., 2014), we exposed the right primary auditory cortex of the mouse (Chen et al., 2011). Cal-520 AM was dissolved in DMSO with 20% Pluronic F-127 to a final concentration of 567 μ M with artificial cerebral spinal fluid (ACSF) for bolus loading. With a standard pipette solution, the solution was diluted 1/10 for cell loading. After that, we used a micropipette filling with the solution and inserted coaxially into the targeted cortex. Finally, the dye-containing solution was ejected in the cortical area by applying a pressure pulse to the pipette. Ca²⁺ imaging was performed approximately 2 hours after dye injection and lasted for up to 8 hours (Stosiek et al., 2003). The mice were trained to perform a sound-evoked licking task (Wang et al., 2020).

The imaging experiment was conducted with a custom-built two-photon microscope system constructed with a 12 kHz resonant scanner (model “LotosScan 1.0,” Suzhou Institute of Biomedical Engineering and Technology, Suzhou, China), in accordance with the set-up described previously (Jia et al., 2010, 2014). Two-photon excitation light was delivered by a mode-locked Ti:Sa laser (model “Mai-Tai DeepSee,” Spectra Physics, CA, United States), and a 40 \times /0.8 NA water-immersion objective (Nikon, Minato, Japan) was used for imaging. For Ca²⁺ imaging experiments, the excitation wavelength was set to 920 nm. For neuronal population imaging, images of 600 \times 600

pixels were acquired at a 40 Hz frame rate. The size of the field-of-view was approximately $200 \times 200 \mu\text{m}$. The average power delivered to the brain was 30–120 mW. For dendritic spine imaging, we used the LOTOS procedure (Chen et al., 2011, 2012). The images, consisting of 250×250 pixels, were acquired at a 160 Hz frame rate. The size of the field-of-view was therefore reduced to approximately $25 \times 25 \mu\text{m}$. The average power delivered to the brain was 15–30 mW.

Motion Correction Method

Motion Correction Pipeline for Two-Photon Imaging Data

Fast image feature extraction and registration (FIFER) operates in a rigid correction flow (Figure 1A). Initially, we generate features for an image with a density-based estimating and clustering algorithm (Figure 1B). This process begins a calculation with the template image (here we used the first frame of imaging data) and the first raw frame for correction (i.e., the second frame of imaging data). After obtaining features of the template and first raw frame, we then generated features for the rest of the raw frames by updating the features obtained from the first raw frame with a fast process (Figure 1C). After we obtained features for each frame, we matched the features between the template image and the raw frames according to the similarity metric (Figure 1D). Finally, we calculated the rigid transformation parameters and conducted the image registration. The computational details of each step are described in the following sections. In the figures, the brightness of the representative two-photon images was adjusted for better visualization.

Generating Features From Two-Photon Image

To perform image density-based feature estimating and clustering, we first initialized the generation of features with the density estimate. To minimize the computational cost, we acquired the coordinates around the high-intensity area, instead of randomly inputting coordinates, to decrease the cost for the following hill climbing procedure. In this work, we assume that the features from accelerated segment test (FAST) (Rosten et al., 2008) detector is a suitable algorithm to detect corners with high speed. Figure 2A demonstrates that the corners detected by the FAST algorithm are mostly distributed in high-intensity areas. To reduce the number of repeated FAST features, we down-sampled these coordinates by merging those near each other (Figure 2B). Based on the initial FAST-detected features, we described the distribution of the empirical density field in an image. Inspired by Denclue (Hinneburg and Keim, 1998), we propose a weighted kernel density estimation (W-KDE) algorithm, with which we mapped each image pixel to a Gaussian kernel function weighted by this pixel's image value as its estimated density:

$$\hat{\mathbf{f}}_{\text{Density}}(\vec{p}) = \frac{1}{h^2 \sum_{i=1}^n \mathbf{I}(\vec{p}_i)} \sum_{i=1}^n \mathbf{I}(\vec{p}_i) \cdot \mathbf{K}_{\text{Gaussian}}\left(\frac{\vec{p} - \vec{p}_i}{h}\right) \quad (1)$$

$$\mathbf{K}_{\text{Gaussian}}(\vec{u}) = (2\pi)^{-1} \cdot e^{-\frac{(\vec{u})^2}{2}}$$

where $\vec{p} = \langle x, y \rangle$ is an arbitrary coordinate in a partitioned ROI (region of interest) of an image, $\vec{p}_i = \langle x_i, y_i \rangle$ is one coordinate

in the ROI, $\mathbf{I}(\vec{p}_i)$ is the pixel value in this position \vec{p}_i , n is the number of pixels sampled from the ROI, h is a bandwidth to smooth the density distribution, and $\mathbf{K}_{\text{Gaussian}}(\vec{u})$ is a Gaussian kernel function which maps a two-dimensional vector \vec{u} onto a constant. Considering the distance between a given coordinate and pixel intensity, we describe the influence from one pixel \vec{p}_i with a pixel value of $\mathbf{I}(\vec{p}_i)$ to an arbitrary coordinate \vec{p} of the image as $\mathbf{I}(\vec{p}_i) \cdot \mathbf{K}_{\text{Gaussian}}\left(\frac{\vec{p} - \vec{p}_i}{h}\right)$, which is quantified by a weighted Gaussian kernel function. Thus, we could estimate the density of an arbitrary coordinate \vec{p} in a partitioned ROI by calculating the sampled ROI's pixel component density and normalizing the sum of these components, which ensures that the whole distribution is a normalized density distribution. In brief, $\hat{\mathbf{f}}_{\text{Density}}(\vec{p})$ is the weighted mixture density of position \vec{p} within the total influence of the ROI. To demonstrate that, we selected a fully sized ROI around an image and calculated each pixel's density given the ROI (Figure 3A) through W-KDE and revealed the density field (Figure 3B) to visualize the density distribution.

Since we estimated a ROI's density distribution, we analyzed its ridge topography for feature extraction. Here, we obtained the coordinates of high intensity regions and hence determined the local maximums as cluster centers (Hinneburg and Keim, 1998). The eq. (2) shows the derivation of W-KDE function with the hill climbing algorithm:

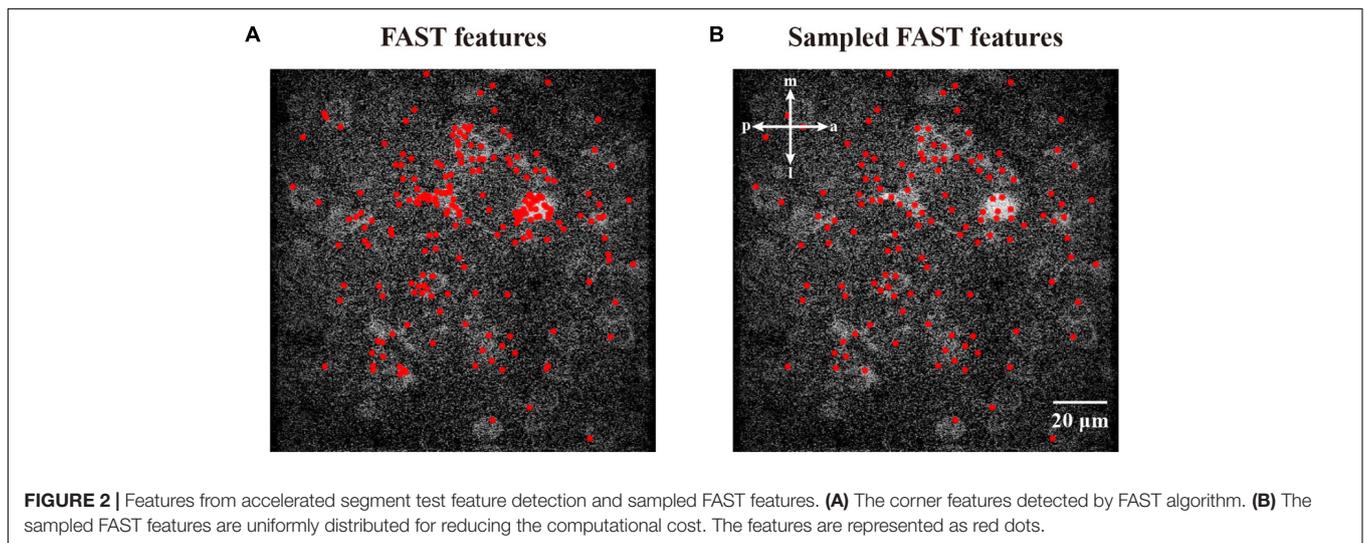
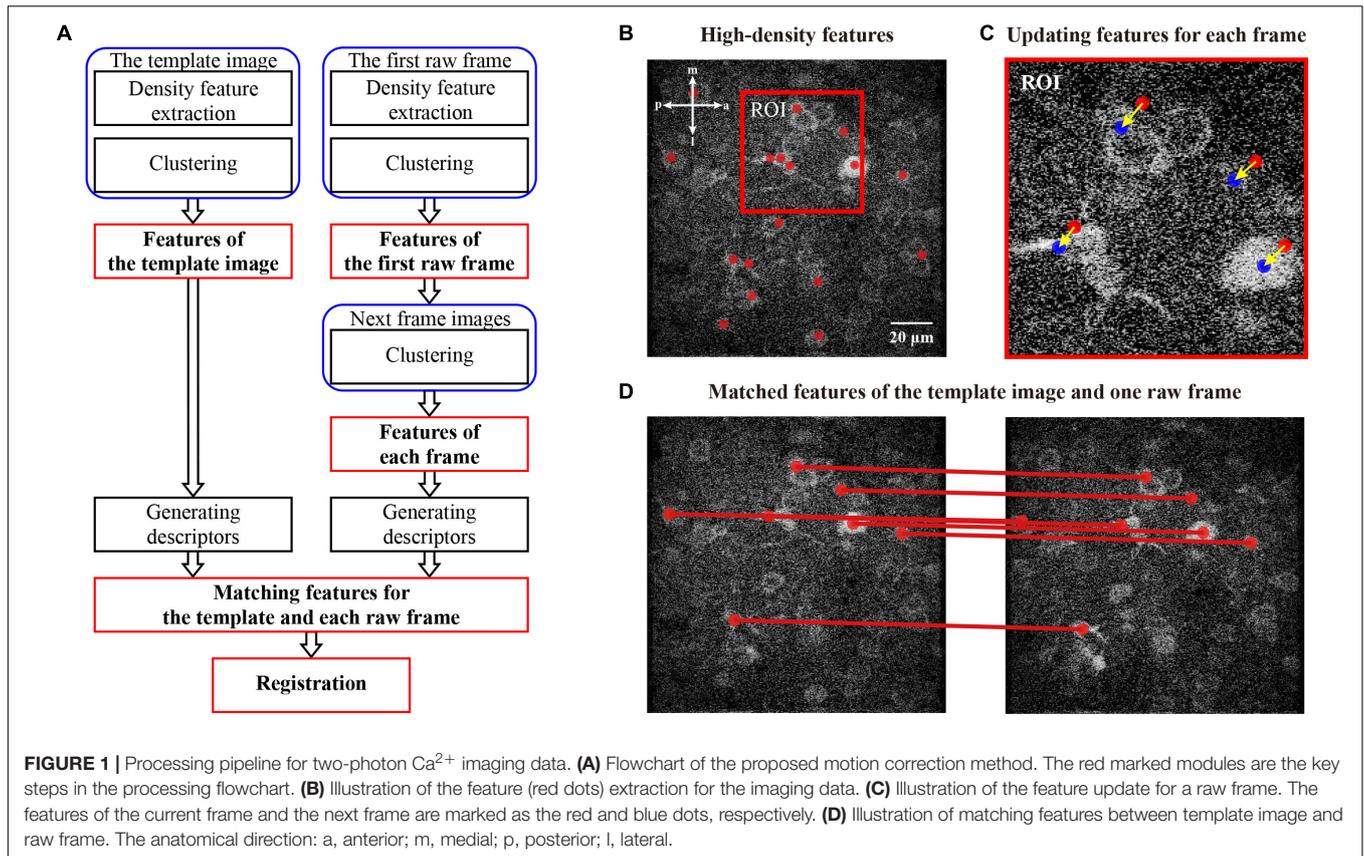
$$\begin{aligned} \nabla \hat{\mathbf{f}}_{\text{Density}}(\vec{p}) &= \frac{1}{h^4 \sum_{i=1}^n \mathbf{I}(\vec{p}_i)} \sum_{i=1}^n \mathbf{I}(\vec{p}_i) \cdot \mathbf{K}_{\text{Gaussian}}\left(\frac{\vec{p} - \vec{p}_i}{h}\right) \\ &\quad \cdot (\vec{p} - \vec{p}_i) \\ \vec{p}^{t+1} &= \vec{p}^t + \delta \frac{\nabla \hat{\mathbf{f}}_{\text{Density}}(\vec{p}^t)}{\|\nabla \hat{\mathbf{f}}_{\text{Density}}(\vec{p}^t)\|_2} \quad (2) \end{aligned}$$

where $\nabla \hat{\mathbf{f}}_{\text{Density}}(\vec{p})$ is the gradient of an arbitrary coordinate \vec{p} in the ROI, and δ is the step stride in hill climbing. Other variables are the same as W-KDE equation. However, a fixed-step hill climbing algorithm is generally poor in efficiency. Therefore, we adapted a fast automatically step-adapted hill climbing approach in our algorithm, which was based on previous work (Hinneburg and Gabriel, 2007), as eq. (3) shows.

$$\begin{aligned} \vec{p}^{t+1} &= \frac{\sum_{i=1}^n \mathbf{I}(\vec{p}_i) \cdot \mathbf{K}_{\text{Gaussian}}\left(\frac{\vec{p}^t - \vec{p}_i}{h}\right) \cdot \vec{p}_i}{\sum_{i=1}^n \mathbf{I}(\vec{p}_i) \cdot \mathbf{K}_{\text{Gaussian}}\left(\frac{\vec{p}^t - \vec{p}_i}{h}\right)} \quad (3) \\ \frac{\hat{\mathbf{f}}_{\text{Density}}(\vec{p}^{t+1}) - \hat{\mathbf{f}}_{\text{Density}}(\vec{p}^t)}{\hat{\mathbf{f}}_{\text{Density}}(\vec{p}^{t+1})} &\leq \varepsilon \end{aligned}$$

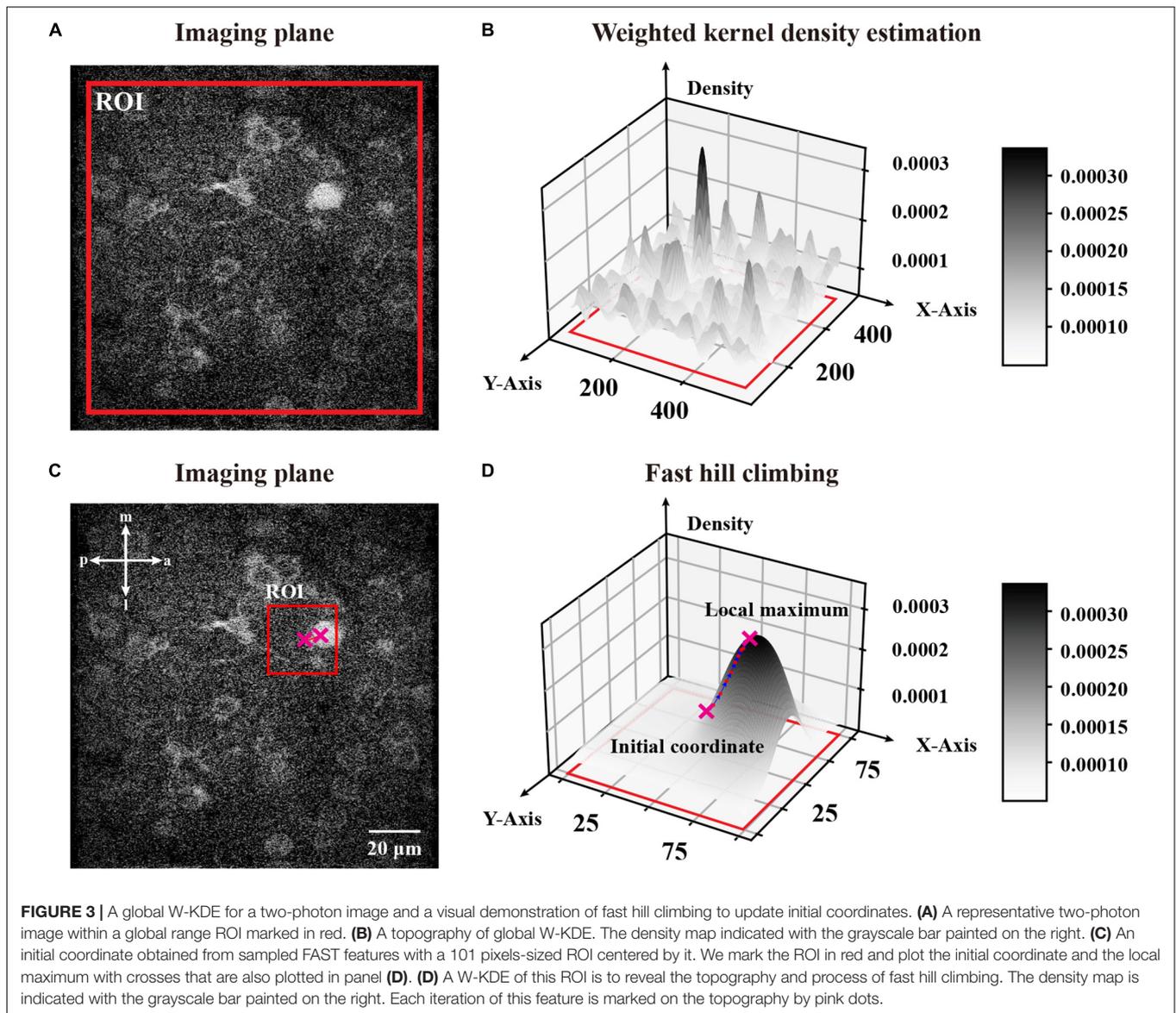
where ε is a given restriction to end hill climbing that also represents the precision of the local maximum. Therefore, we optimized the hill climbing to a faster algorithm by reducing the iteration numbers based on the increase in rate of density. Alternatively, we can simply stop the iteration when a coordinate update distance is smaller than a threshold η .

$$\|\vec{p}^{t+1} - \vec{p}^t\|_2 \leq \eta \quad (4)$$



Hence we can start with any coordinate in the ROI and cluster it to a local maximum, which is the top of a ridge topography and probably represents a neuronal object. To update the initial coordinates to a local maximum, we used a ROI centered by the feature with an appropriate size of 101 pixels (Figure 3C). After repeating the fast hill climbing procedure (Figure 3D) to the coordinates of sampled FAST features with a rough termination, where we calculate step distance of each feature during iterations

and simply terminate the fast hill climbing when the step distance is smaller than a rough constant η in eq. (4), and we obtained rough features with noise (Figure 4A). Then, we sorted all of the rough features by their estimated densities and treated the upper quartile features as filtered results (Figure 4B). Finally, we repeated the fast hill climbing procedure on rough features with a precise termination, where we finally terminate the hill climbing when the density change rate of each feature is smaller than a



critical value as described in eq. (3), and merged the coincident features (**Figure 4C**). After conducting these processes on the template image and the first raw frame individually, we obtained the initial pair of image features.

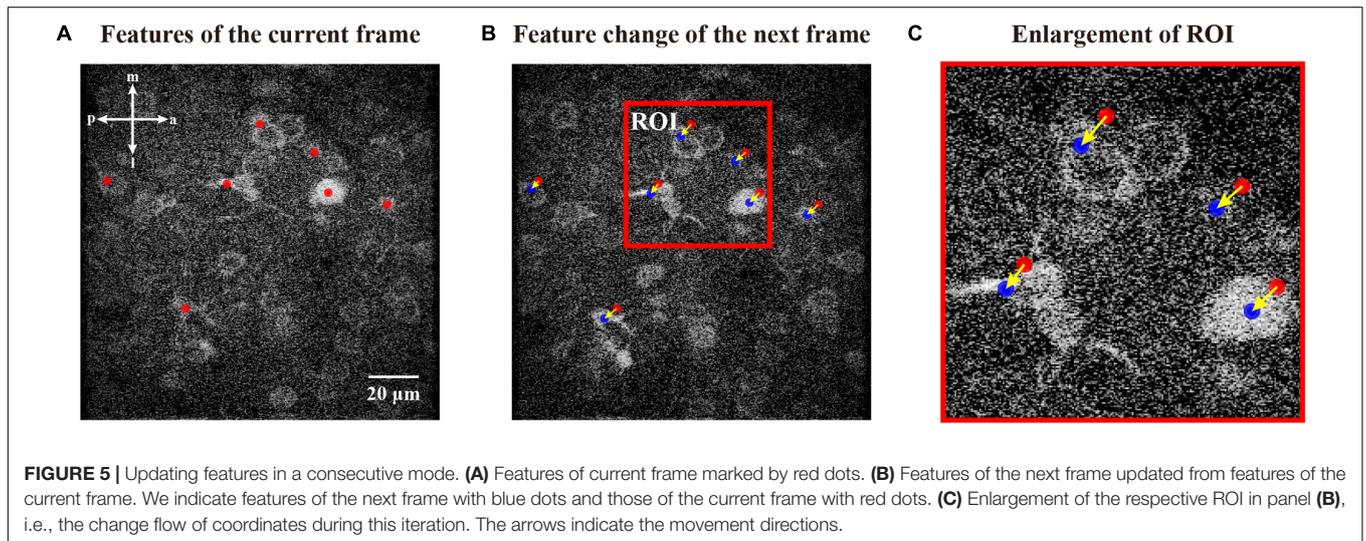
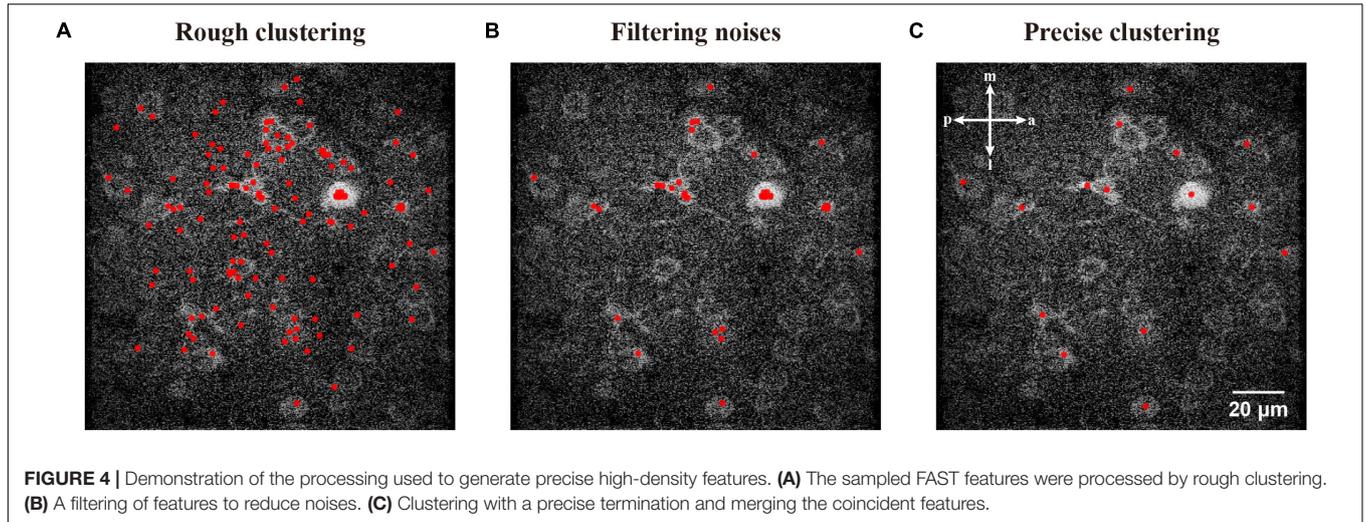
Updating Features With an Efficient Mode

Starting the calculation for the features of the first raw frame, we updated the features of the current frame to those of the next frame through the following procedure (**Figure 5**). First, we used a fixed-size ROI centered on every coordinate of the feature in the current frame under the hypothesis that the features between two consecutive frames are adjacent. Then, we applied the fast hill climbing algorithm to achieve local maxima in the same way as before. This procedure is ultrafast and efficient for updating features from consecutive imaging frames due to: (1) Only a small number of features, which were extracted from the first raw frame, were updated in this process; (2) features between two

consecutive frames were normally adjacent, so the hill climbing iteration was stopped quickly; and (3) the fast hill climbing algorithm could automatically adapt the step of hill climbing depending on the distance to the local maximum. Therefore, our approach has an advantage in computational efficiency by updating features from consecutive imaging frames.

Matching Corresponding Features

When we obtained features of the template image (**Figure 6A**) and one raw frame in the imaging sequence (**Figure 6B**), we matched those features to obtain transformation parameters for image registration. We then constructed the feature vector in three dimensions: the x - and y -axis positions and the normalized density (**Figures 6C,D**). In addition, we assumed that the template image and the raw frame have similar image structure in terms of position and density. Based on this hypothesis, we deduced that the coupled features in the template image



and raw frame have the same relative position and estimated density. Thus, we propose to create a descriptor for each feature by reserving the differences between this feature and the rest (**Figures 6E,F**).

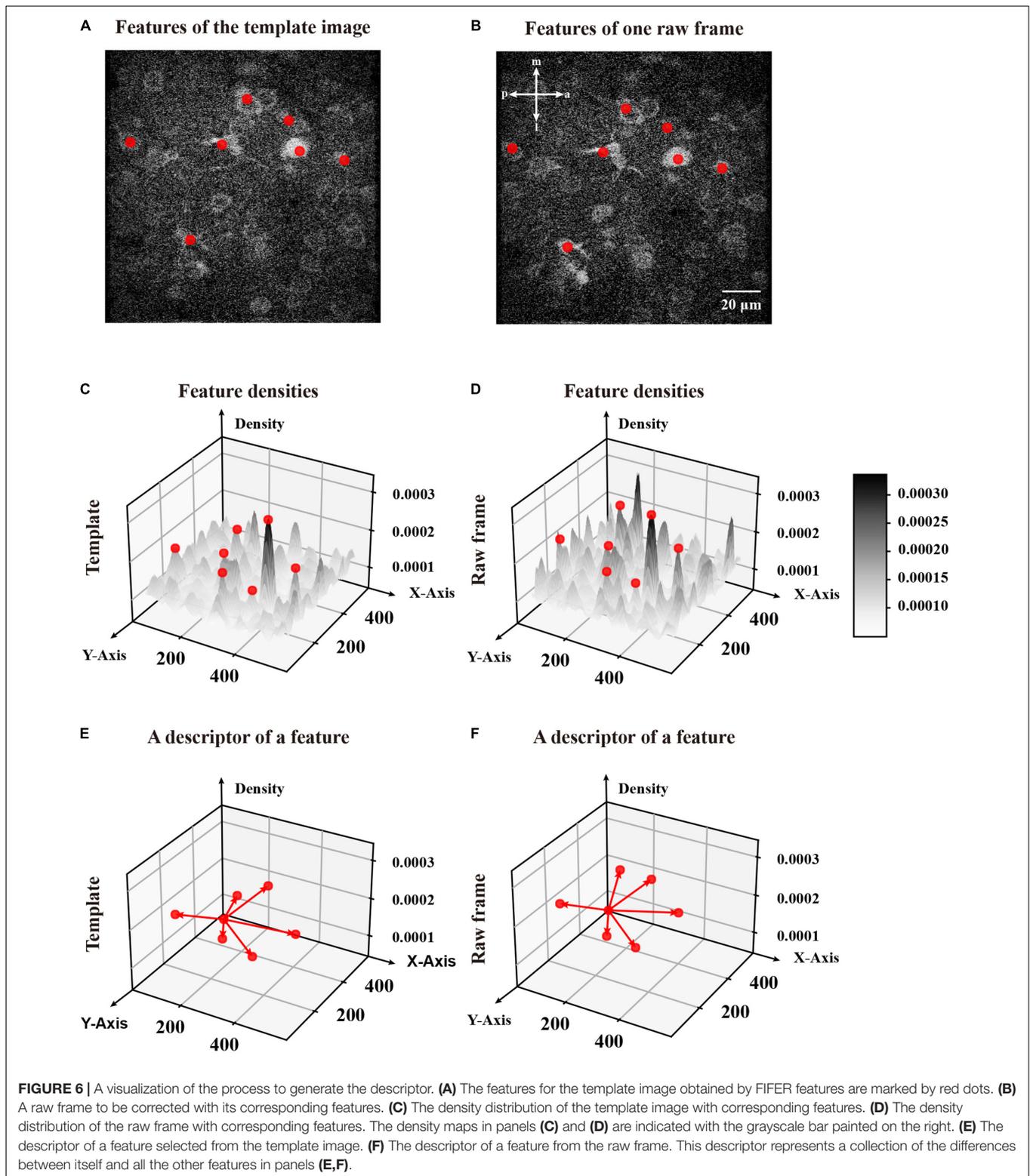
$$\left\{ \begin{array}{l} P_{Template} = [\vec{p}_1^{Temp}, \vec{p}_2^{Temp}, \dots, \vec{p}_{k_1-1}^{Temp}, \vec{p}_{k_1}^{Temp}] \\ P_{Target} = [\vec{p}_1^{Targ}, \vec{p}_2^{Targ}, \dots, \vec{p}_{k_2-1}^{Targ}, \vec{p}_{k_2}^{Targ}] \\ Descriptor(\vec{p}_\alpha^{Temp}) = [\vec{u}_1, \vec{u}_2, \dots, \vec{u}_{k_1-1}, \vec{u}_{k_1}] \\ Descriptor(\vec{p}_\beta^{Targ}) = [\vec{v}_1, \vec{v}_2, \dots, \vec{v}_{k_2-1}, \vec{v}_{k_2}] \\ \vec{u}_{k_1} = \vec{p}_{k_1}^{Temp} - \vec{p}_\alpha^{Temp} \\ \vec{v}_{k_2} = \vec{p}_{k_2}^{Targ} - \vec{p}_\beta^{Targ} \end{array} \right. \quad (5)$$

where the descriptors \vec{p}_α^{Temp} or \vec{p}_β^{Targ} are in their groups of point sets. After obtaining the descriptor of each feature, we matched the features by comparing the similarity of their descriptors. To match two feature descriptors, the corresponding elements between two descriptors were calculated for the Jaccard similarity coefficient with a parameter ζ_1 set as the tolerable

limit. If the calculated Jaccard similarity coefficient was larger than a given threshold ζ_2 , their corresponding two features were successfully matched. Eq. (6) describes this matching process.

$$\left\{ \begin{array}{l} \frac{J(Descriptor(\vec{p}_\alpha^{Temp}), Descriptor(\vec{p}_\beta^{Targ}))}{Descriptor(\vec{p}_\alpha^{Temp}) \cap Descriptor(\vec{p}_\beta^{Targ})} \\ = \frac{Descriptor(\vec{p}_\alpha^{Temp}) \cup Descriptor(\vec{p}_\beta^{Targ})}{\tau} \\ = \frac{2(k_1 + k_2 - \tau)}{\tau} \quad (6) \\ \tau = \sum_{m,n=1}^{k_1, k_2} L\left(\frac{2\|\vec{u}_m - \vec{v}_n\|_2}{\|\vec{u}_m\|_2 + \|\vec{v}_n\|_2}\right) - \tau_{same} \\ L(\sigma) = \begin{cases} 1, \sigma \leq \zeta_1 \\ 0, \sigma > \zeta_1 \end{cases} \end{array} \right.$$

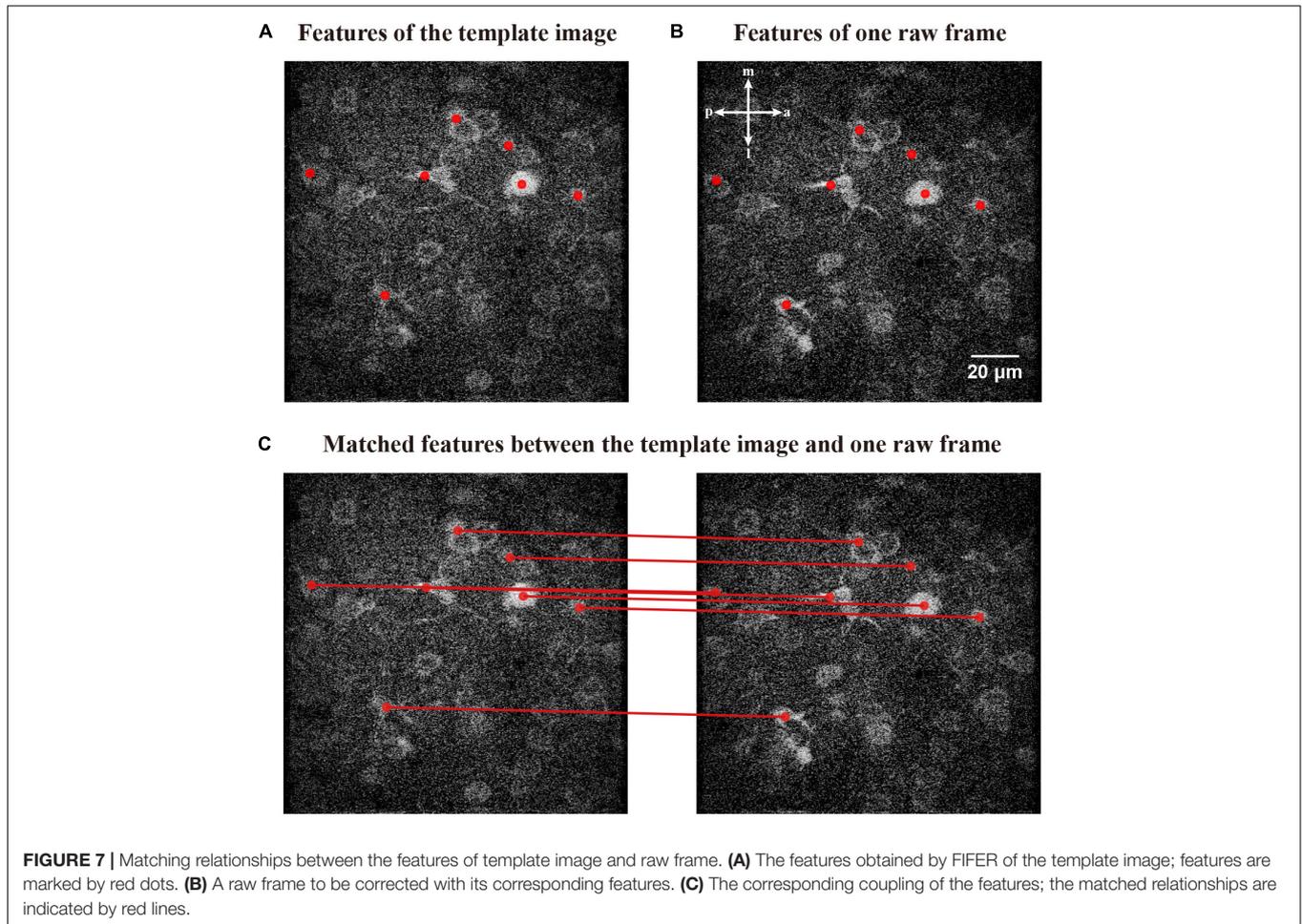
where τ_{same} represents the counts of multi-matching features. We demonstrate matching the features of a template image



(Figure 7A) with a raw frame (Figure 7B). The matched relations of those features are shown in Figure 7C.

In addition, we calculated the rotation transformation parameters for the image feature descriptors. Inspired by SIFT

(Lowe, 2004), we used a histogram of oriented gradients (HoG) to set a constructive main direction to transform each feature for matching images (Figures 8A,B). Initially, we created several bins to calculate the distributions of the summed magnitude of vectors



within 36 directions for each descriptor. Then, we decomposed each element of a descriptor of magnitudes into the upper and lower boundaries and counted the total magnitude in those 36 directions (**Figures 8C,D**). After that, we determined the main direction whose bin had the largest total magnitude. To obtain continuous directions, we used parabola interpolation to estimate a relative precise direction. Since we obtained the individual direction of a descriptor, we transformed the descriptor for matching features (**Figures 8E,F**). Then, we estimated a rotation angle by applying singular value decomposition (SVD) to multiple couples of the matched features (Besl and McKay, 1992). Hence we calculated the covariance matrix among two groups of features and applied SVD to obtain a transformation matrix and perform the registration of the image.

Evaluation of Motion Correction Performance

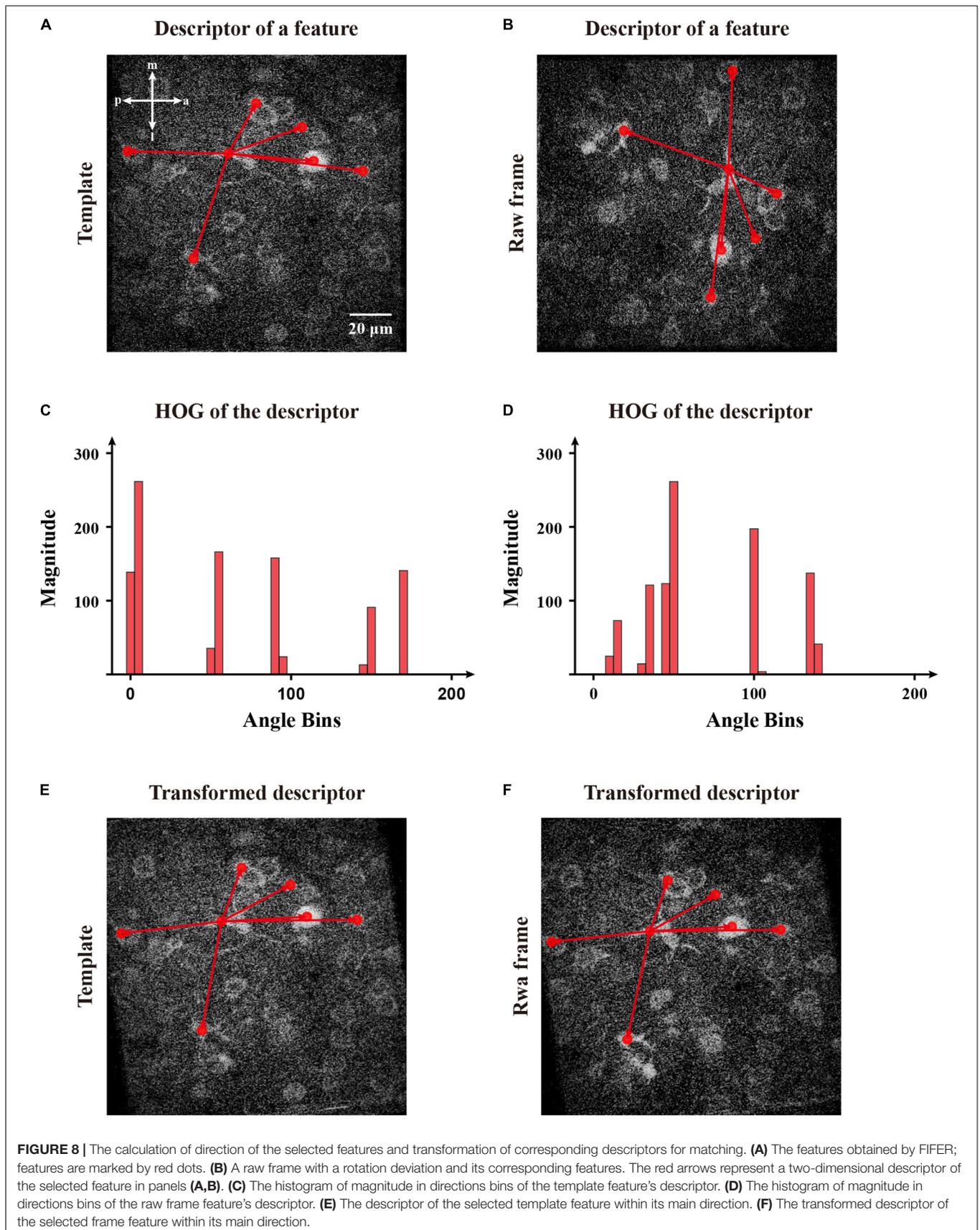
For the acquired two-photon imaging datasets, the similarities between the template image and corrected imaging data frames were calculated to evaluate motion correction performance. Here, four popular image quality metrics, including the normalized root mean square error (NRMSE), the peak signal-to-noise ratio (PSNR), the structural similarity (SSIM) index, and

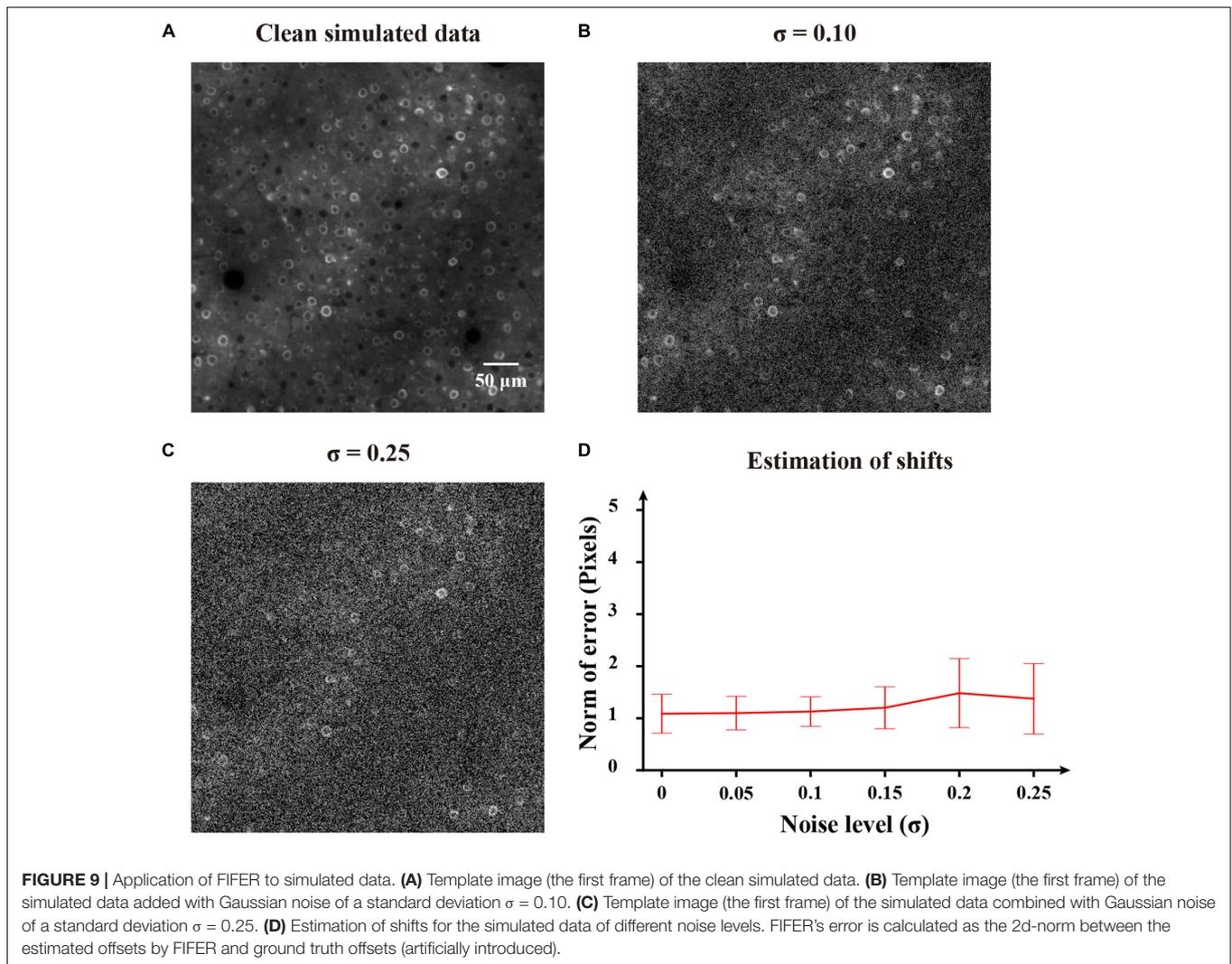
the normalized mutual information (NMI), were adopted in our evaluation tests. A smaller NRMSE value and a larger PSNR/SSIM/NMI value indicate better image correction quality (Luo et al., 2021). The calculation time was measured for processing each raw two-photon image of 600×600 , 512×512 (neuronal population imaging) or 250×250 pixels (dendritic spine imaging) using an AMD Ryzen 7 5800X 3.8 GHz CPU, with 32 GB RAM.

RESULTS

Validation of Motion Correction by Fast Image Feature Extraction and Registration

To first perform an experimental validation of the proposed method, we evaluated the motion correction results on simulated imaging data. Here we used the NAOMi simulation toolbox (Song et al., 2021) to simulate two-photon imaging data. We set the numerical aperture (NA) of the objective lens as 0.8, the NA of the illumination light as 0.6 and the laser power as 40 mW. The generated data video had a size of 500×500 pixels and 100 frames. **Figure 9A** shows a representative example of





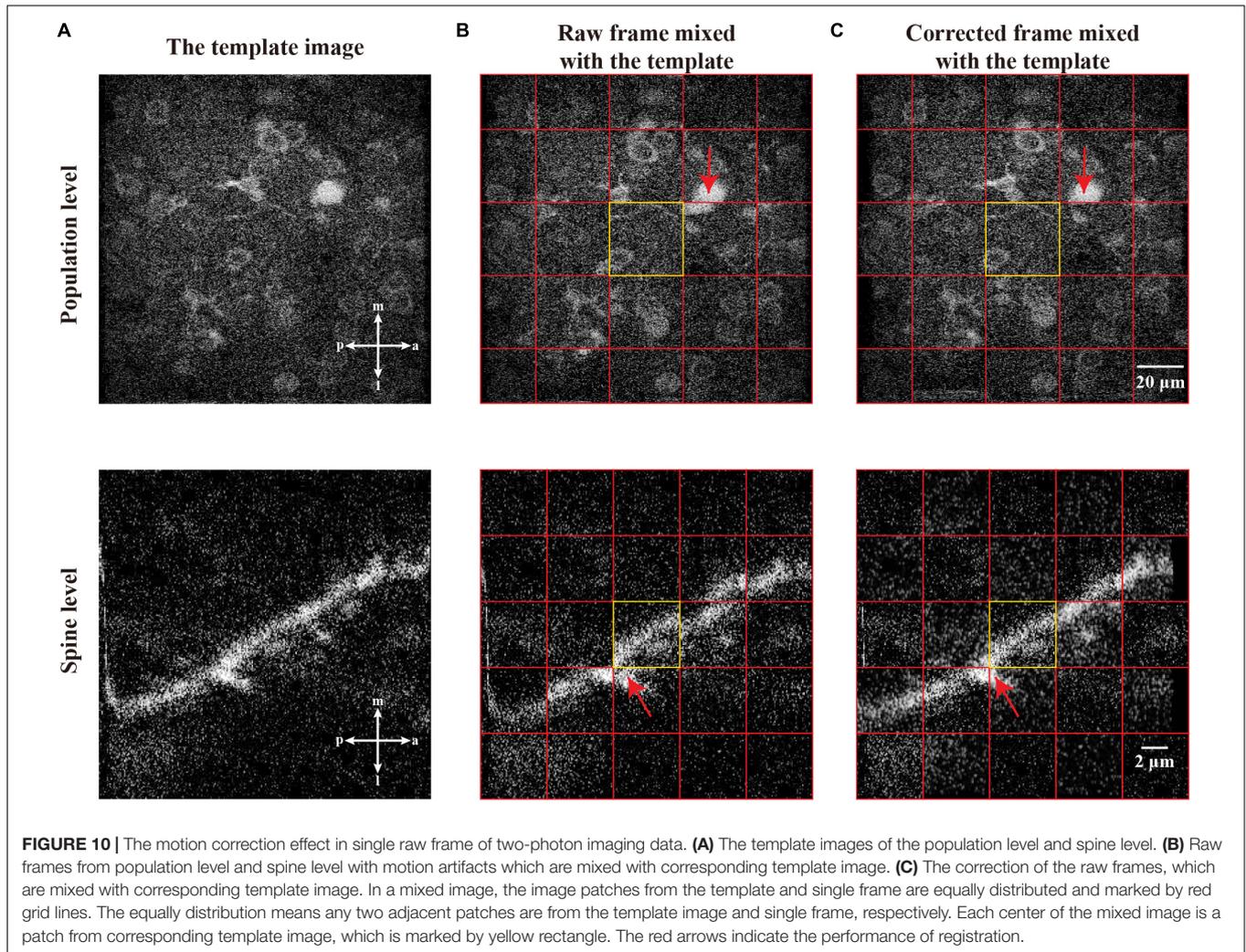
simulated clean two-photon image. Then the synthetic noisy imaging data were generated with random spatial shifts and added Gaussian noise as the previous work (Pnevmatikakis and Giovannucci, 2017). Gaussian noise with zero mean and standard deviation sampled from $[0, 0.25]$ were added to the clean two-photon images to generate testing images. **Figures 9B,C** show two noisy shifted images (noise levels: $\sigma = 0.10$ and $\sigma = 0.25$). As we can see (**Figure 9D**), the FIFER algorithm can estimate the shifts remarkably well. We evaluated the performance of motion correction for the simulated data with different noise levels, and the estimate errors do not change much with the noise level increases. Therefore, our proposed method achieved good performance on the simulation data test.

To further evaluate the performance of motion correction, we used two different two-photon imaging scales of datasets, i.e., the scale of a neuronal population (**Figure 10**, top) and the scale of a dendritic spine (**Figure 10**, bottom). **Figures 10A,B** shows the examples of the template images and the raw imaging frames (mixed with template images) to be corrected, respectively. After we applied our motion correction method for processing the

raw imaging frames, the two representative frames were aligned to their corresponding templates (**Figure 10C**). This figure demonstrates that the raw frames were correctly transformed (here, a translation) to be matched with the template images.

In **Figure 11**, we demonstrate the motion correction effects by showing the average image of the imaging data before and after motion correction. As the motion artifacts distort some of the imaging frame sequences, the average of a functional imaging data might suffer from inconsistency. As can be seen from the averaged raw image of the two imaging datasets (**Figure 11A**), both the neuronal population and dendritic spine imaging data show a blur effect due to motion artifacts during recording. By contrast, after we performed our motion correction method with the raw imaging data, the average of corrected images shows greatly improved quality, and the cells and spines are clearly visible without obvious blurred effect (**Figure 11B**). Thus, the imaging data were successfully restored.

We further addressed the effects of motion corrected neuronal activity as a time series of fluorescence changes. We identified the individual neurons and extracted their Ca²⁺ signals from the



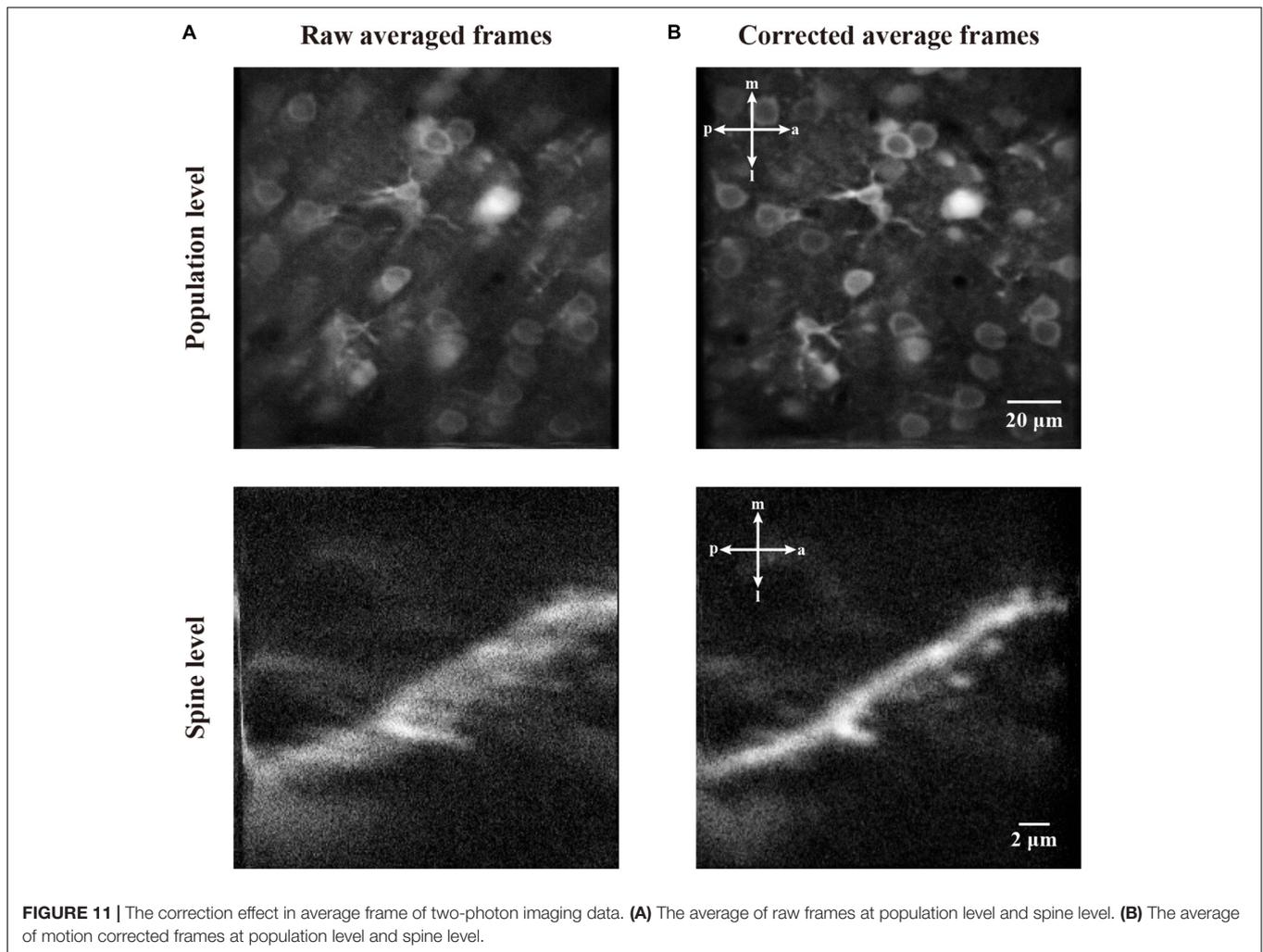
imaging frames to assess the change dynamics of Ca²⁺ transients both before (**Figure 12A**) and after (**Figure 12B**) motion correction. The raw imaging data exhibit that the motion resulted in spike-like changes (**Figure 12C**, left) or even larger distortions (**Figure 12D**, left) in the time series, which are indicated by the black arrows in **Figures 12C,D**. After we processed the raw imaging data with our method, the motion artifact-induced changes were successfully reduced and the neuronal signals were clearly restored (**Figures 12C,D**, right). Hence, the quality of individual Ca²⁺ transients processed by motion correction algorithm was clearly higher than that of the raw signals, which can facilitate downstream analyses, e.g., the detection of soma (Guan et al., 2018) and neuronal Ca²⁺ transients.

Comparison of Different Methods for Motion Correction

To further assess the motion correction performance of our proposed method, we compared FIFER with other popular image registration methods. We tested them with three two-photon imaging datasets, including neuronal population ($n = 200$ and

$n = 1825$) and dendritic spine ($n = 1500$) imaging frames to evaluate the motion correction performance of the methods. The image registration methods used for comparison consisted of two groups: feature-based methods and intensity-based methods. For the classic feature-based methods, the SIFT (Lowe, 2004), ORB (Rublee et al., 2011), and AKAZE (Alcantarilla et al., 2013) methods were used for comparison. For the widely used intensity-based methods, TurboReg (Thevenaz et al., 1998), Moco (Dubbs et al., 2016), NoRMCorre (Pnevmatikakis and Giovannucci, 2017), the real-time processing method (Mitani and Komiyama, 2018), and Suite2p (Pachitariu et al., 2017) were used for testing.

First, we tested FIFER and the abovementioned methods with our dataset of neuronal population imaging. As the testing results showing in **Table 1**, our FIFER method exhibits superior motion correction performance compared to existing image registration approaches in terms of NRMSE (0.9131 ± 0.0416), PSNR (19.7529 ± 0.3930), SSIM (0.1972 ± 0.0105), and NMI (0.0281 ± 0.0027). In addition, the calculation time of our proposed method is just 2.92 ms for processing each image. For the feature-based methods, SIFT shows better relative

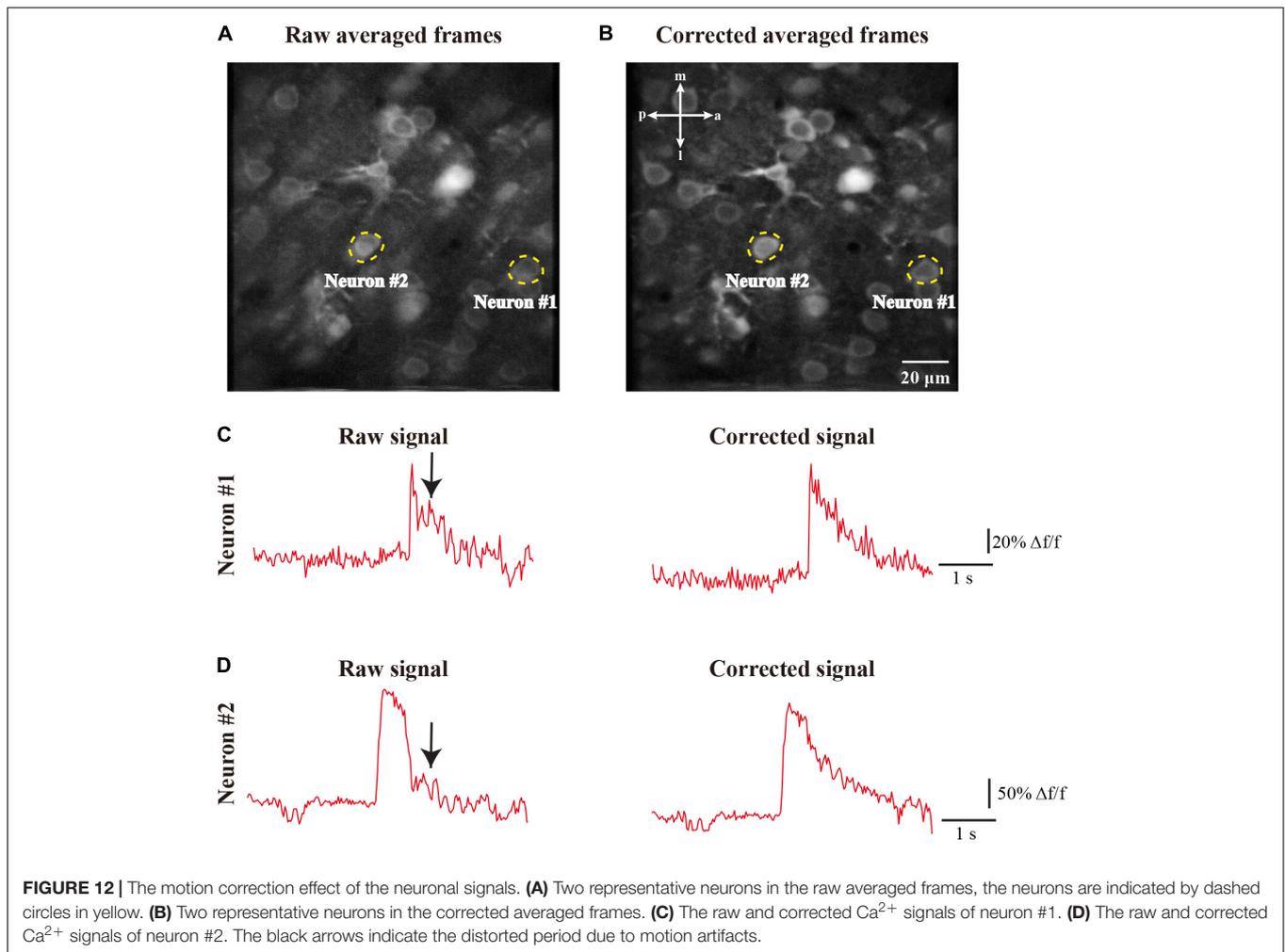


motion correction performance than ORB and AKAZE with a longer computational time, and ORB is relatively fast but has poor correction ability. For the intensity-based methods, we tested TurboReg with two working modes: Accurate and Fast. The Accurate mode of TurboReg achieved a better correction performance than that of Fast mode, while also requiring a longer calculation time. As the previous work reported, Moco achieved a fast processing of the data, using 26.92 ms for processing each frame. However, the motion correction performance of Moco is just comparable to the Fast mode of TurboReg. Moreover, we also tested NoRMCorre with its two modes, Rigid and Non-rigid. The Non-rigid mode of NoRMCorre performed well and was the closest to matching the motion correction performance achieved by FIFER, ranking second in our comparative analysis. However, the calculation time for the Non-rigid mode was the longest among the tested methods. In comparison, the Rigid mode of NoRMCorre also achieved good results with a much shorter time for processing. For the real-time processing method, it was very fast when processing the raw frames, requiring 3.85 ms for each frame, and its correction performance was as the same level as that of Moco and the Fast mode of TurboReg. For correcting the

images with Suite2p, it used 11.91 ms for processing each frame and revealed similar performance as that of Moco. Hence Suite2p achieved a moderate motion correction accuracy and speed. Taken together, our proposed method not only provides superior motion correction results ($P < 0.05$, paired t -test), but also uses minimum processing time compared to the other tested methods.

We next compared FIFER with those methods for another dendritic spine imaging dataset, because this is also a popular imaging scale. As the results (**Table 2**) demonstrate, our proposed method also provided the best correction results ($P < 0.05$, paired t -test) in terms of NRMSE (0.8490 ± 0.0508), PSNR (18.5548 ± 0.5045), SSIM (0.1791 ± 0.0149), NMI (0.0733 ± 0.0043), and the fastest processing speed (0.59 ms for processing each frame). Among the other tested methods, the Accurate mode of TurboReg achieved the second-best correction result, and the real-time processing method achieved the second fastest processing speed (0.76 ms for processing each frame).

To supplement the validation datasets, we further tested FIFER and those methods with a publicly available dataset provided by the CaImAn project (Giovannucci et al., 2019). The testing results (**Table 3**) show that, FIFER (1.77 ms for processing



each frame) was clearly faster than all other algorithms, meanwhile it also achieved the best ($P < 0.05$, paired t -test) correction performance in terms of NRMSE (0.7930 ± 0.1346), PSNR (27.0004 ± 1.3278), SSIM (0.4632 ± 0.0382), and NMI (0.0434 ± 0.0033). These validation results indicate a high generalization ability of FIFER, as it performed consistently fast and accurate for processing multiple datasets acquired from different labs.

DISCUSSION

In this work, we proposed the FIFER method to perform motion correction for two-photon Ca²⁺ imaging data by extracting features *via* image density-based estimation and raw image registration. Using a regular personal computer, the proposed method showed promising performance in both computational speed and motion correction precision. The testing results showed that our method achieved strong results for not only simulated imaging data (Figure 9) but also both neuronal population and dendritic spine imaging datasets (Figures 10, 11). In addition, we demonstrated that the individual neuronal signal

quality was clearly improved using our approach (Figure 12). The comparative analyses of FIFER against previously reported image registration approaches for motion correction in three imaging datasets (Tables 1–3) highlight the superior ability of our proposed method, as it is faster than other approaches while also achieving the best image correction accuracy. Hence, adopting our image correction method for online two-photon imaging experiments will benefit both functional imaging and closed loop photostimulation. The code implementing FIFER is published on the project's GitHub page.

To extract features for matching images, we proposed a new algorithm, namely, W-KDE and density-based clustering, to extract the density features from a single frame of two-photon imaging data. Our proposed method outperformed the other feature-based and intensity-based methods for motion correction because it operates in a robust way to extract features, and then finds the optimal template matching solution. As our proposed method registered a pair of images by extracting local features, it reduced the differential contents from the global image. By contrast, the other feature-based methods, including SIFT, ORB, and AKAZE, did not show good performance. The reason for this might be that their feature extraction suffered from the

TABLE 1 | Comparison of fast image feature extraction and registration (FIFER) with other methods (Mean \pm SD) for neuronal population imaging dataset ($n = 200$ frames, 600×600 pixels for each frame).

Method	Metric (<i>P</i> -value: FIFER vs. other method)				Time (ms)
	NRMSE	PSNR	SSIM	NMI	
SIFT	0.9922 \pm 0.0641 (<i>P</i> = 1.12e-35)	19.0407 \pm 0.5627 (<i>P</i> = 6.13e-36)	0.1726 \pm 0.0224 (<i>P</i> = 2.46e-35)	0.0217 \pm 0.0059 (<i>P</i> = 7.30e-35)	105.23
ORB	1.0202 \pm 0.0566 (<i>P</i> = 2.91e-57)	18.7942 \pm 0.4903 (<i>P</i> = 2.00e-57)	0.1594 \pm 0.0248 (<i>P</i> = 1.42e-51)	0.0191 \pm 0.0065 (<i>P</i> = 3.15e-45)	26.87
AKAZE	1.0082 \pm 0.0703 (<i>P</i> = 8.81e-45)	18.9045 \pm 0.6131 (<i>P</i> = 6.41e-45)	0.1740 \pm 0.0181 (<i>P</i> = 1.10e-41)	0.0223 \pm 0.0045 (<i>P</i> = 2.22e-41)	52.41
TurboReg (Accurate)	0.9738 \pm 0.0345 (<i>P</i> = 2.59e-42)	19.1906 \pm 0.3063 (<i>P</i> = 1.93e-42)	0.1799 \pm 0.0124 (<i>P</i> = 1.68e-55)	0.0236 \pm 0.0012 (<i>P</i> = 2.78e-69)	140.87
TurboReg (Fast)	1.0369 \pm 0.0212 (<i>P</i> = 2.46e-96)	18.6414 \pm 0.1759 (<i>P</i> = 8.20e-94)	0.1700 \pm 0.0110 (<i>P</i> = 4.53e-92)	0.0223 \pm 0.0009 (<i>P</i> = 1.23e-85)	118.15
Moco	1.0361 \pm 0.0211 (<i>P</i> = 9.23e-96)	18.6484 \pm 0.1749 (<i>P</i> = 2.96e-93)	0.1715 \pm 0.0109 (<i>P</i> = 3.41e-86)	0.0224 \pm 0.0009 (<i>P</i> = 6.30e-84)	26.92
NoRMCorre (Rigid)	0.9601 \pm 0.0422 (<i>P</i> = 9.29e-23)	19.3160 \pm 0.3796 (<i>P</i> = 4.60e-23)	0.1842 \pm 0.0119 (<i>P</i> = 2.17e-34)	0.0240 \pm 0.0014 (<i>P</i> = 1.46e-56)	61.28
NoRMCorre (Non-rigid)	0.9521 \pm 0.0181 (<i>P</i> = 1.46e-28)	19.3817 \pm 0.1641 (<i>P</i> = 5.53e-29)	0.1922 \pm 0.0090 (<i>P</i> = 1.28e-12)	0.0246 \pm 0.0009 (<i>P</i> = 6.73e-53)	210.21
Real-time processing	1.0147 \pm 0.0667 (<i>P</i> = 7.10e-49)	18.8815 \pm 1.2811 (<i>P</i> = 4.70e-18)	0.1755 \pm 0.0591 (<i>P</i> = 2.93e-7)	0.0254 \pm 0.0270 (<i>P</i> = 1.55e-1)	3.85
Suite2p	1.0087 \pm 0.0254 (<i>P</i> = 3.99e-73)	18.8821 \pm 0.2166 (<i>P</i> = 3.11e-72)	0.1694 \pm 0.0130 (<i>P</i> = 5.86e-82)	0.0226 \pm 0.0013 (<i>P</i> = 1.06e-81)	11.91
FIFER	0.9131 \pm 0.0416	19.7529 \pm 0.3930	0.1972 \pm 0.0105	0.0281 \pm 0.0027	2.92

The calculation time is for each frame. Statistical tests were calculated using the paired *t*-test.

TABLE 2 | Comparison of FIFER with other methods (mean \pm SD) for dendritic spine imaging dataset ($n = 1500$ frames, 250×250 pixels for each frame).

Method	Metric (<i>P</i> -value: FIFER vs. other method)				Time (ms)
	NRMSE	PSNR	SSIM	NMI	
SIFT	0.9253 \pm 0.0815 (<i>P</i> = 7.06e-260)	17.8256 \pm 0.7608 (<i>P</i> = 9.59e-264)	0.1657 \pm 0.0192 (<i>P</i> = 2.78e-108)	0.0661 \pm 0.0089 (<i>P</i> = 1.50e-254)	15.79
ORB	0.9344 \pm 0.0783 (<i>P</i> = 1.11e-313)	17.7370 \pm 0.7262 (<i>P</i> = 2.47e-319)	0.1630 \pm 0.0191 (<i>P</i> = 2.32e-145)	0.0647 \pm 0.0090 (<i>P</i> = 6.61e-295)	10.92
AKAZE	0.9312 \pm 0.0809 (<i>P</i> = 1.90e-294)	17.7694 \pm 0.7513 (<i>P</i> = 2.95e-299)	0.1628 \pm 0.0202 (<i>P</i> = 7.10e-49)	0.0668 \pm 0.0077 (<i>P</i> = 3.73e-280)	7.78
TurboReg (Accurate)	0.9064 \pm 0.0292 (<i>P</i> = 1.45e-258)	17.9752 \pm 0.2775 (<i>P</i> = 1.59e-264)	0.1539 \pm 0.0090 (<i>P</i> = 9.99e-140)	0.0719 \pm 0.0027 (<i>P</i> = 2.27e-76)	109.58
TurboReg (Fast)	0.9654 \pm 0.0135 (<i>P</i> = 0)	17.4240 \pm 0.1197 (<i>P</i> = 0)	0.1391 \pm 0.0045 (<i>P</i> = 0)	0.0711 \pm 0.0025 (<i>P</i> = 2.70e-154)	104.46
Moco	0.9604 \pm 0.0111 (<i>P</i> = 0)	17.4687 \pm 0.1000 (<i>P</i> = 0)	0.1411 \pm 0.0043 (<i>P</i> = 0)	0.0715 \pm 0.0025 (<i>P</i> = 1.80e-104)	9.24
NoRMCorre (Rigid)	0.9325 \pm 0.0189 (<i>P</i> = 0)	17.7263 \pm 0.1742 (<i>P</i> = 0)	0.1462 \pm 0.0054 (<i>P</i> = 0)	0.0711 \pm 0.0025 (<i>P</i> = 9.10e-175)	7.59
NoRMCorre (Non-rigid)	0.9301 \pm 0.0147 (<i>P</i> = 0)	17.7480 \pm 0.1361 (<i>P</i> = 0)	0.1485 \pm 0.0049 (<i>P</i> = 0)	0.0720 \pm 0.0026 (<i>P</i> = 5.10e-77)	57.4
Real-time processing	0.9550 \pm 0.0254 (<i>P</i> = 0)	17.5269 \pm 0.5620 (<i>P</i> = 0)	0.1417 \pm 0.0222 (<i>P</i> = 0)	0.0718 \pm 0.0096 (<i>P</i> = 5.54e-9)	0.76
Suite2p	0.9506 \pm 0.0252 (<i>P</i> = 0)	17.5606 \pm 0.2262 (<i>P</i> = 0)	0.1428 \pm 0.0061 (<i>P</i> = 0)	0.0697 \pm 0.0029 (<i>P</i> = 4.00e-323)	1.90
FIFER	0.8490 \pm 0.0508	18.5548 \pm 0.5045	0.1791 \pm 0.0149	0.0733 \pm 0.0043	0.59

The calculation time is for each frame. Statistical tests were calculated using the paired *t*-test.

signal-to-noise ratio issues with the imaging data. The insufficient common neuronal morphology or uneven background might cause classical feature-based motion correction methods to

fail. Specially, the features extracted by our approach mostly represent some certain cellular structures: the feature marks the strongest Ca²⁺ signals of the soma or the dendritic

TABLE 3 | Comparison of FIFER with other methods (Mean \pm SD) for neuronal population imaging dataset (the CalmAn dataset file images_N.01.01, $n = 1825$ frames, 512×512 pixels for each frame).

Method	Metric (<i>P</i> -value: FIFER vs. other method)				Time (ms)
	NRMSE	PSNR	SSIM	NMI	
SIFT	0.9611 \pm 0.1576 (<i>P</i> = 0)	25.3388 \pm 1.4258 (<i>P</i> = 0)	0.3193 \pm 0.1376 (<i>P</i> = 3.48e-295)	0.0218 \pm 0.0161 (<i>P</i> = 0)	67.54
ORB	0.9523 \pm 0.1563 (<i>P</i> = 0)	25.4124 \pm 1.3676 (<i>P</i> = 0)	0.3744 \pm 0.0696 (<i>P</i> = 0)	0.0283 \pm 0.0097 (<i>P</i> = 0)	18.31
AKAZE	0.8843 \pm 0.1436 (<i>P</i> = 0)	26.0484 \pm 1.2999 (<i>P</i> = 0)	0.4022 \pm 0.0433 (<i>P</i> = 0)	0.0346 \pm 0.0039 (<i>P</i> = 0)	40.28
TurboReg (Accurate)	0.8789 \pm 0.1365 (<i>P</i> = 0)	26.0914 \pm 1.2283 (<i>P</i> = 0)	0.4093 \pm 0.0367 (<i>P</i> = 0)	0.0347 \pm 0.0025 (<i>P</i> = 0)	112.07
TurboReg (Fast)	0.8943 \pm 0.1398 (<i>P</i> = 0)	25.9418 \pm 1.2370 (<i>P</i> = 0)	0.4005 \pm 0.0383 (<i>P</i> = 0)	0.0337 \pm 0.0024 (<i>P</i> = 0)	109.20
Moco	0.8945 \pm 0.1402 (<i>P</i> = 0)	25.9403 \pm 1.2400 (<i>P</i> = 0)	0.4003 \pm 0.0386 (<i>P</i> = 0)	0.0337 \pm 0.0024 (<i>P</i> = 0)	15.00
NoRMCorre (Rigid)	0.8899 \pm 0.1396 (<i>P</i> = 0)	25.9848 \pm 1.2412 (<i>P</i> = 0)	0.4027 \pm 0.0383 (<i>P</i> = 0)	0.0340 \pm 0.0024 (<i>P</i> = 0)	39.77
NoRMCorre (Non-rigid)	0.8821 \pm 0.1392 (<i>P</i> = 0)	26.0631 \pm 1.2486 (<i>P</i> = 0)	0.4090 \pm 0.0379 (<i>P</i> = 0)	0.0346 \pm 0.0024 (<i>P</i> = 0)	264.38
Real-time processing	0.8897 \pm 0.1390 (<i>P</i> = 0)	25.9805 \pm 1.2984 (<i>P</i> = 0)	0.3976 \pm 0.0401 (<i>P</i> = 0)	0.0342 \pm 0.0119 (<i>P</i> = 1.70e-179)	2.70
Suite2p	0.8881 \pm 0.1400 (<i>P</i> = 0)	26.0037 \pm 1.2457 (<i>P</i> = 0)	0.4041 \pm 0.0385 (<i>P</i> = 0)	0.0341 \pm 0.0024 (<i>P</i> = 0)	8.28
FIFER	0.7930 \pm 0.1346	27.0004 \pm 1.3278	0.4632 \pm 0.0382	0.0434 \pm 0.0033	1.77

The calculation time is for each frame. Statistical tests were calculated using the paired *t*-test.

spine, hence the extracted features by our approach are more accurate than the traditional feature extraction approaches for two-photon images. In addition, the intensity-based methods, including TurboReg, Moco, NoRMCorre, Suite2p and the real-time processing method, also perform image registration based on global image information. Hence, they might also encounter the issue that the intensity difference between the non-common areas (e.g., the noisy background) from the image pair may have a negative impact on the image registration and result in a sub-optimal image registration solution (Li et al., 2020). Furthermore, the sparse features extracted from foreground cellular structures by our approach hardly suffer from the background noise and might lead to an optimal solution. For instance, the dendritic spine imaging data only presented small common areas, so FIFER using local features achieved better results than other methods (Table 2). Therefore, the testing results show that our feature extraction and matching method provides robust results for correcting motion artifacts at different two-photon imaging scales.

Moreover, the testing results show that our method is ultrafast to find an optimal matching solution for an image pair. Only the real-time processing method achieved comparable calculation speed, and the other methods are much slower (Tables 1–3). With our algorithm, once features were extracted for the first raw frame, we could conduct an ultrafast update of features with nearby ROI gradients within several iterations. Across the whole imaging frame sequence, this kind of feature extraction strategy needs only once global image search to cluster features for the first raw frame and

template image. Following this initial search, the features of the remaining raw frames can be quickly generated based on the previous information. By contrast, the other methods used global image content (Mitani and Komiyama, 2018); as a result, their computational costs were relatively larger than that of our method.

It is also worth to note that, the features in the raw two-photon images are non-stationary, hence one of the best approaches for feature detection might be performing spatio-temporal feature extraction to tackle temporal inconsistencies. Although computational complexity increases with spatio-temporal information extracted from consecutive frames of imaging data, the processing can preserve both the morphology features and temporal neural dynamics (Luo et al., 2021). As another consequence of non-stationarity, it also poses a challenge for evaluating motion correction, because single metric is hardly able to quantify the performance well. Therefore, a series of complementary metrics in both spatial and temporal dimensions can be combined as the most appropriate solution for measuring the performance quantification, which still needs further investigations.

In this study, we demonstrate that FIFER performed well for motion correction task and required only a conventional computer, such success indicates that it would be desirable to implement FIFER as online processing software and use it for neurobiology experiments requiring online observation and stimulation. However, it warrants to mention that FIFER is designed as a frame-by-frame 2D motion correction method, hence it has the limitation for dealing with the distortions

induced by 3D brain movements, which requires 3D motion correction method, e.g., (Griffiths et al., 2020). In addition, although our method demonstrated promising motion correction performance and a fast processing speed in a rigid mode, it would be worthwhile to extend our method to treat affine (Li et al., 2020) or non-rigid distortions (Pnevmatikakis and Giovannucci, 2017) in complex imaging experiments, such as in chronic two-photon imaging to study learning-induced neuronal activity changes.

DATA AVAILABILITY STATEMENT

The imaging data supporting the conclusions of this article are available from the corresponding author upon reasonable request. The code is provided at <https://github.com/Weiyi-Liu-Unique/FIFER>. We used a public dataset (the CaImAn dataset file images_N.01.01.zip from <https://zenodo.org/record/1659149>) to evaluate the performance of FIFER and other methods. For a detailed description of the dataset (see Giovannucci et al., 2019).

ETHICS STATEMENT

This study was approved by the Institutional Animal Care and Use Committee of Third Military Medical University. All experimental procedures were conducted in accordance with

animal ethical guidelines of the Third Military Medical University Animal Care and Use Committee.

AUTHOR CONTRIBUTIONS

KZ, XC, XnL, and XaL contributed to the design of the study and wrote the manuscript with help from other authors. JP and MW performed the imaging experiments and acquired the data. WL, YX, XnL, and XaL designed the method. WL, JP, YX, HJ, KZ, and XC processed the data sets. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported by grants from the National Natural Science Foundation of China (No. 32171096) to XaL and the National Natural Science Foundation of China (Nos. 31925018, 31861143038, and 32127801) to XC. XC was a junior fellow of the CAS Center for Excellence in Brain Science and Intelligence Technology.

ACKNOWLEDGMENTS

We thank Jia Lou for technical assistance.

REFERENCES

- Alcantarilla, P. F., Nuevo, J., and Bartoli, A. (2013). "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *Proceedings of the British Machine Vision Conference (BMVC)*, Bristol, 1–11. doi: 10.5244/C.27.13
- Besl, P. J., and McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal.* 14, 239–256. doi: 10.1109/34.121791
- Chen, J. L., Pfäffli, O. A., Voigt, F. F., Margolis, D. J., and Helmchen, F. (2013). Online correction of licking-induced brain motion during two-photon imaging with a tunable lens. *J. Physiol.* 591, 4689–4698. doi: 10.1113/jphysiol.2013.259804
- Chen, X., Leischner, U., Rochefort, N. L., Nelken, I., and Konnerth, A. (2011). Functional mapping of single spines in cortical neurons in vivo. *Nature* 475, 501–505. doi: 10.1038/nature10193
- Chen, X., Leischner, U., Varga, Z., Jia, H., Deca, D., Rochefort, N. L., et al. (2012). LOTOS-based two-photon calcium imaging of dendritic spines in vivo. *Nat. Protoc.* 7, 1818–1829. doi: 10.1038/nprot.2012.106
- Dombeck, D. A., Khabbaz, A. N., Collman, F., Adelman, T. L., and Tank, D. W. (2007). Imaging large-scale neural activity with cellular resolution in awake, mobile mice. *Neuron* 56, 43–57. doi: 10.1016/j.neuron.2007.08.003
- Dubbs, A., Guevara, J., and Rafael, Y. (2016). Moco: fast motion correction for calcium imaging. *Front. Neuroinform.* 10:6. doi: 10.3389/fninf.2016.00006
- Giovannucci, A., Friedrich, J., Gunn, P., Kalfon, J., Brown, B. L., Koay, S. A., et al. (2019). CaImAn an open source tool for scalable calcium imaging data analysis. *elife* 8:e38173. doi: 10.7554/eLife.38173
- Greenberg, D. S., and Kerr, J. N. (2009). Automated correction of fast motion artifacts for two-photon imaging of awake animals. *J. Neurosci. Meth.* 176, 1–15. doi: 10.1016/j.jneumeth.2008.08.020
- Grienberger, C., and Konnerth, A. (2012). Imaging calcium in neurons. *Neuron* 73, 862–885. doi: 10.1016/j.neuron.2012.02.011
- Griffiths, V. A., Valera, A. M., Lau, J. Y., Roš, H., Younts, T. J., Marin, B., et al. (2020). Real-time 3D movement correction for two-photon imaging in behaving animals. *Nat. Methods* 17, 741–748. doi: 10.1038/s41592-020-0851-7
- Guan, J., Li, J., Liang, S., Li, R., Li, X., Shi, X., et al. (2018). NeuroSeg: automated cell detection and segmentation for in vivo two-photon Ca²⁺ imaging data. *Brain Struct. Funct.* 223, 519–533. doi: 10.1007/s00429-017-1545-5
- Haskins, G., Kruger, U., and Yan, P. (2020). Deep learning in medical image registration: a survey. *Mach. Vision Appl.* 31: 8. doi: 10.1007/s00138-020-01060-x
- Hinneburg, A., and Gabriel, H.-H. (2007). "Denclue 2.0: fast clustering based on kernel density estimation," in *Advances in Intelligent Data Analysis VII. IDA 2007. Lecture Notes in Computer Science*, Vol. 4723, eds R. M. Berthold, J. Shawe-Taylor, and N. Lavraè (Berlin: Springer), 70–80. doi: 10.1007/978-3-540-74825-0_7
- Hinneburg, A., and Keim, D. A. (1998). An efficient approach to clustering in large multimedia databases with noise. *Proc. KDD* 98, 58–65. doi: 10.1155/2018/9391635
- Huber, D., Gutnisky, D. A., Peron, S., O'Connor, D. H., Wiegert, J. S., Tian, L., et al. (2012). Multiple dynamic representations in the motor cortex during sensorimotor learning. *Nature* 484, 473–478. doi: 10.1038/nature11039
- Jia, H., Rochefort, N. L., Chen, X., and Konnerth, A. (2010). Dendritic organization of sensory input to cortical neurons in vivo. *Nature* 464, 1307–1312. doi: 10.1038/nature08947
- Jia, H., Varga, Z., Sakmann, B., and Konnerth, A. (2014). Linear integration of spine Ca²⁺ signals in layer 4 cortical neurons in vivo. *Proc. Natl. Acad. Sci. U.S.A.* 111, 9277–9282. doi: 10.1073/pnas.1408525111
- Li, C., Yang, X., Ke, Y., and Yung, W. H. (2020). Fully affine invariant methods for cross-session registration of calcium imaging data. *eNeuro* 7:ENEURO.0054-0020.2020. doi: 10.1523/eneuro.0054-20.2020
- Li, R., Wang, M., Yao, J., Liang, S., Liao, X., Yang, M., et al. (2018). Two-photon functional imaging of the auditory cortex in behaving mice: from neural networks to single spines. *Front. Neural Circuits* 12:33. doi: 10.3389/fncir.2018.00033
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* 60, 91–110. doi: 10.1023/B:VISI.0000029664.99615.94
- Luo, L., Xu, Y., Pan, J., Wang, M., Guan, J., Liang, S., et al. (2021). Restoration of two-photon Ca²⁺ imaging data through model blind

- spatiotemporal filtering. *Front. Neurosci.* 15:630250. doi: 10.3389/fnins.2021.630250
- Mitani, A., and Komiyama, T. (2018). Real-time processing of two-photon calcium imaging data including lateral motion artifact correction. *Front. Neuroinform.* 12:98. doi: 10.3389/fninf.2018.00098
- Pachitariu, M., Stringer, C., Dipoppa, M., Schröder, S., Rossi, L. F., Dalglish, H., et al. (2017). Suite2p: beyond 10,000 neurons with standard two-photon microscopy. *bioRxiv* [Preprint]. doi: 10.1101/061507
- Pneumatikakis, E. A. (2019). Analysis pipelines for calcium imaging data. *Curr. Opin. Neurobiol.* 55, 15–21. doi: 10.1016/j.conb.2018.11.004
- Pneumatikakis, E. A., and Giovannucci, A. (2017). NoRMCorre: an online algorithm for piecewise rigid motion correction of calcium imaging data. *J. Neurosci. Methods* 291, 83–94. doi: 10.1016/j.jneumeth.2017.07.031
- Rosten, E., Porter, R., and Drummond, T. (2008). Faster and better: a machine learning approach to corner detection. *IEEE Trans. Pattern Anal.* 32, 105–119. doi: 10.1109/TPAMI.2008.275
- Ruble, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). “ORB: an efficient alternative to SIFT or SURF” in *Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV)*, Barcelona, 2564–2571. doi: 10.1109/ICCV.2011.6126544
- Song, A., Gauthier, J. L., Pillow, J. W., Tank, D. W., and Charles, A. S. (2021). Neural anatomy and optical microscopy (NAOMi) simulation for evaluating calcium imaging methods. *J. Neurosci. Methods* 358:109173. doi: 10.1016/j.jneumeth.2021.109173
- Stosiek, C., Garaschuk, O., Holthoff, K., and Konnerth, A. (2003). In vivo two-photon calcium imaging of neuronal networks. *Proc. Natl. Acad. Sci. U.S.A.* 100, 7319–7324. doi: 10.1073/pnas.1232232100
- Tada, M., Takeuchi, A., Hashizume, M., Kitamura, K., and Kano, M. (2014). A highly sensitive fluorescent indicator dye for calcium imaging of neural activity in vitro and in vivo. *Eur. J. Neurosci.* 39, 1720–1728. doi: 10.1111/ejn.12476
- Thevenaz, P., Ruttimann, U. E., and Unser, M. (1998). A pyramid approach to subpixel registration based on intensity. *IEEE Trans. Image Process.* 7, 27–41. doi: 10.1109/83.650848
- Thompson, N. C., Greenewald, K., Lee, K., and Manso, G. F. (2020). The computational limits of deep learning. *arXiv* [Preprint]. arXiv: 2007.05558,
- Wang, M., Liao, X., Li, R., Liang, S., Ding, R., Li, J., et al. (2020). Single-neuron representation of learned complex sounds in the auditory cortex. *Nat. Commun.* 11:4361. doi: 10.1038/s41467-020-18142-z
- Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
- Publisher’s Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.
- Copyright © 2022 Liu, Pan, Xu, Wang, Jia, Zhang, Chen, Li and Liao. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.