Video Article

Application of Automated Image-guided Patch Clamp for the Study of Neurons in Brain Slices

Qiuyu Wu[1,2], Alexander A. Chubykin[1,2]

[1]Department of Biological Sciences, Purdue University

[2]Purdue Institute for Integrative Neuroscience, Purdue University

Correspondence to: Alexander A. Chubykin at chubykin@purdue.edu

URL: https://www.jove.com/video/56010
DOI: doi:10.3791/56010

Keywords: Neuroscience, Issue 125, Automatic patching, patch clamp, *in vitro* electrophysiology, computer vision, fluorescent cell detection, Python

Date Published: 7/31/2017

Citation: Wu, Q., Chubykin, A.A. Application of Automated Image-guided Patch Clamp for the Study of Neurons in Brain Slices. *J. Vis. Exp.* (125), e56010, doi:10.3791/56010 (2017).

Abstract

Whole-cell patch clamp is the gold-standard method to measure the electrical properties of single cells. However, the *in vitro* patch clamp remains a challenging and low-throughput technique due to its complexity and high reliance on user operation and control. This manuscript demonstrates an image-guided automatic patch clamp system for *in vitro* whole-cell patch clamp experiments in acute brain slices. Our system implements a computer vision-based algorithm to detect fluorescently labeled cells and to target them for fully automatic patching using a micromanipulator and internal pipette pressure control. The entire process is highly automated, with minimal requirements for human intervention. Real-time experimental information, including electrical resistance and internal pipette pressure, are documented electronically for future analysis and for optimization to different cell types. Although our system is described in the context of acute brain slice recordings, it can also be applied to the automated image-guided patch clamp of dissociated neurons, organotypic slice cultures, and other non-neuronal cell types.

Video Link

The video component of this article can be found at https://www.jove.com/video/56010/

Introduction

The patch clamp technique was first developed by Neher and Sakmann in the 1970s to study the ionic channels of excitable membranes[1]. Since then, patch clamping has been applied to the study of many different subjects at the cellular, synaptic, and circuit level—both *in vitro* and *in vivo*—in many different cell types, including neurons, cardiomyocytes, Xenopus oocytes, and artificial liposomes[2]. This process involves the correct identification and targeting of a cell of interest, intricate micromanipulator control to move the patch pipette in close proximity to the cell, the application of positive and negative pressure to the pipette at the proper time to establish a tight gigaseal patch, and a break-in to establish a whole-cell patch configuration. Patch clamping is typically conducted manually and requires extensive training to master. Even for a researcher experienced with the patch clamp, the success rate is relatively low. More recently, several attempts have been made to automate patch-clamp experiments. Two main strategies have evolved to accomplish automation: augmenting standard patch clamp equipment to provide automatic control of the patching process and the design of new equipment and techniques from the ground up. The former strategy is adaptable to existing hardware and can be used in a variety of patch clamp applications, including *in vivo* blind patch clamp[3,4,5], *in vitro* patch clamp of acute brain slices, organotypic slice cultures, and cultured dissociated neurons[6]. It enables the interrogation of complex local circuits by using multiple micromanipulators simultaneously[7]. The planar patch method is an example of the new development strategy, which can achieve the high-throughput simultaneous patch clamp of cells in suspension for drug screening purposes[8]. However, the planar patch method is not applicable to all cell types, particularly neurons with long processes or intact circuits containing extensive connections. This limits its application to mapping the intricate circuitry of the nervous system, which is a key advantage of traditional patch clamp technology.

We have developed a system that automates the manual patch clamp process *in vitro* by augmenting standard patch clamp hardware. Our system, Autopatcher IG, provides automatic pipette calibration, fluorescent cell target identification, automatic control of pipette movement, automatic whole-cell patching, and data logging. The system can automatically acquire multiple images of brain slices at different depths; analyze them using computer vision; and extract information, including the coordinates of fluorescently labeled cells. This information can then be used to target and automatically patch cells of interest. The software is written in Python—a free, open-source programming language—using several open-source libraries. This ensures its accessibility to other researchers and improves the reproducibility and rigor of electrophysiology experiments. The system has a modular design, such that additional hardware can easily be interfaced with the current system demonstrated here.

Protocol

1. System Setup

1. Construct the pressure control unit.
    1. Assemble the pressure control unit according to the circuit map (**Figure 1**). Solder the necessary parts onto the Printed Circuit Board (PCB) manufactured according to the electrical circuit schematics (**Figure 1b**). Use standard resistors, LEDs, Metal-Oxide Semiconductor Field-Eeffect Transistors (MOSFETs), capacitors, and connectors (see the **Table of Materials**). Solder solenoid valves onto the PCB. Connect the air pump and air pressure sensor to the PCB with electrical wire.
    NOTE: It should take about 2 h to construct the pressure control unit with all necessary parts made available.

2. Connect the secondary data acquisition (DAQ) board.
    1. Connect data outputs from the printed circuit board to the DAQ board, following **Table 1**.
    NOTE: The DAQ board will be running in "Single-ended mode." The port map can be found in the user manual (see the **Table of Materials**).
    2. Connect "AIn Pr S" to one of the analog input (AI) channels and "R-Gr" to one of the analog grounds on the secondary DAQ board.
    3. Connect the primary output from the amplifier to one of the AI channels and the ground to the analog ground of the secondary DAQ board.
    NOTE: A standard BNC cable can be used to connect the primary output from the amplifier.
    4. Strip the other end and connect the positive signal (*i.e.* copper core) to the AI channel and the ground (*i.e.* the thin wire around the core) to the analog ground. Repeat this step for a second channel if more than one patch channel is used.
    NOTE: The analog input to the DAQ board will be configured in later steps.
    5. Connect power to the power output of the secondary DAQ board. Use a separate 12 V power source for the pump.

3. Connect the tubing.
    1. Connect the air pump and the two valves according to **Table 2**. Use a 3-way connector to connect the soft tubing from the valve 2 top port, the pressure sensor, and the pipette holder in the last step.
    2. Add another 3-way connector to the tubing connected to the pipette holder if two pipettes are used. Manually switch between the valves and the pipettes in use when patching.

4. Install Autopatcher IG.
NOTE: System requirement: Autopatcher IG was only tested on a PC running Windows 7. It has not been validated for other operating systems. The described procedure applies specifically to the hardware listed in the Table of Materials.
    1. Download Autopatcher-IG from GitHub (https://github.com/chubykin/AutoPatcher_IG).
    2. Install Python (see the **Table of Materials** for the version and download address).
    3. Uninstall the PyQt4 library by typing "pip uninstall PyQt4" in a command line terminal.
    NOTE: The system uses an older version of the PyQt4 library to achieve compatibility with the Qwt and Opencv libraries.
    4. Install Python libraries from historic wheel files (http://www.lfd.uci.edu/~gohlke/pythonlibs/). Find the following files: Numpy (pymc-2.3.6-cp27-cp27m-win32.whl), Opencv (opencv_python-2.4.13.2-cp27-cp27m-win32.whl), Pyqt (PyQt4-4.11.4-cp27-none-win32.whl), and Qwt (PyQwt-5.2.1-cp27-none-win32.whl).
        1. To install the wheel files, go to the directory where the files are saved and type "pip install ***wheelfilename***.whl." Substitute "***wheelfilename***" with the actual name of the file.
        NOTE: "cp27" in the wheel file name indicates Python 2.7 and "win32" indicated Windows 32-bit. If "win32" does not work, try "win64."
    5. To control the CCD camera, download and install the installer for 64-bit (https://www.qimaging.com/support/software/). Then download MicroManager for 64-bit (https://micro-manager.org/wiki/Download_Micro-Manager_Latest_Release) to control the camera in Python.
    6. To control the manipulators and the microscope stage, install control software provided by the manufacturer.
    NOTE: By doing this, the driver necessary to control the manipulators is also installed. The installation package is commonly provided in a CD-ROM.
    7. To control the secondary DAQ board, install the Universal Library from CD-ROM, provided with the purchase of the DAQ board.

5. Configure the hardware for Autopatcher IG.
    1. Connect the microscope stage and manipulator controllers to the computer via USB ports.
    2. Assign COM port numbers to unit 0: microscope stage, unit 1: left manipulator, and unit 2: right manipulator, in this order, in the "ports.csv" configuration file in the "configuration" folder. Leave the other parameters in the ports.csv file (*i.e.* "SCI" and "1") unchanged.
    NOTE: The COM port number information can be found by running the manipulator configuration software provided by the manufacturer. Go to the "settings" tab, select "settings" and the "Motion" page, and read the labels for each tab at the top. Alternatively, this information can be found in the PC Device Manager.
    3. Assign analog input channel numbers on the DAQ board for a pressure sensor and patch channel 1 and 2 (corresponding to unit 1 and 2). Enter the channel number in the "DAQchannels.csv" file in the "configuration" folder.
    NOTE: It is recommended to open the .csv files with the Notepad application instead of a spreadsheet, as it may alter the information when saving changes.

6. Run Autopatcher IG.
    1. Turn on the amplifier, microscope controller, and manipulator controller. Ensure that the amplifier software is running.
    2. Run Autopatcher IG with Python from a command line terminal as follows: first, change the directory (command "cd" for most common terminals) where Autopatcher IG is installed, type "python Autopatcher_IG.pyw" in the command line terminal, and hit the "Enter" key.

NOTE: Do not run the manipulator control software before running Autopatcher IG because it will occupy the microscope stage and manipulator, causing Autopatcher IG to be unable to find the hardware. Manipulator control software can be run after Autopatcher IG is fully initiated if there are additional modules to be controlled (*e.g.,* the inline heater).

7. Calibrate the primary pipette.
    1. Pull patch pipettes as described previously[9]. Fill a pulled glass pipette with internal solution and load it onto the head stage.
    NOTE: Empty glass pipettes have different contrasts under the microscope and may lead to inaccurate calibration.
    2. Move the pipette tip to the microscope visual field and bring it into focus. If the dial pad is used to move the manipulators and/or microscope stage, update the coordinates by pressing "z" on the keyboard.
    NOTE: This action is not necessary if the keyboard (microscope stage: A/D - x-axis, W/S - y-axis, R/F - z-axis; manipulators: H/K - x-axis, U/J - y-axis, O/L - z-axis, 1/2 - unit number) is used to control movement because the coordinates will be updated in real time.
    3. Click the "Start calibration" button on the main Graphic User Interface (GUI) for the corresponding unit on which the pipette is loaded (**Figure 2**).
    NOTE: A pop-up window will appear when the calibration is finished.
    NOTE: Calibration will be carried out automatically, which will take about 3.5 min. Clicking on the same button (switched now to "cancel calibration" after initiating calibration) will abort the calibration attempt.
    4. Save the calibration by clicking "save calibration" at the bottom of the main GUI (it saves the current calibration for both manipulators and can be loaded in the future).
    NOTE: The field of view under low (4 or 10X) and high (40X) magnification must be aligned for secondary calibration to function properly. Please refer to the user manual of the optical system in use for the alignment procedures.

2. Automatic Patch Clamp Procedure

1. Prepare acute brain slices, as described previously[10].
2. Prepare glass pipettes for the patch clamp.
3. Place one brain slice in the center of the recording chamber. Stabilize the slice with a slice hold-down, or "harp."
4. Detect the fluorescent cell.
    1. Find the area of interest under the 4X lens. Move the microscope stage by turning on click-to-move ("CTM") mode and click the center of the area of interest. Alternatively, use the keypad to move the microscope stage (A/D - x-axis, W/S - y-axis, R/F - z-axis).
    2. Switch to the high-magnification lens and adjust the focus by moving the microscope in the z-axis, using R/F on the keypad.
    NOTE: It is recommended to adjust the water bath level so that the focal plane under the low- and high-magnification lenses are the same or similar in the z-axis.
    3. Click the "Detect Cell" button on the main GUI column, "Unit 0." If the LED or laser light source of the setup cannot be controlled by the TTL signal, manually turn on the LED/laser; a pop-up window will appear when the cell detection is finished.
        1. Turn off the LED/laser if necessary; a list of cell coordinates will appear in the "Memory positions" GUI. Remove undesired cells from the list by clicking the "X" button next to the coordinates.
        2. Alternatively, if target cells are not fluorescently labeled, click "Mouse mode" on the main GUI. Click on the cell of interest; a yellow dot with a number will appear on the cell, and the coordinates of the cell will appear in the "Memory positions" GUI.

5. Calibrate the secondary pipette.
    1. Fill 1/3 of a glass pipette with internal solution. Load the pipette onto the pipette holder attached to the head stage.
    2. Use the low-magnification lens. Bring the pipette into the visual field and adjust the focus using the keypad (H/K - x-axis, U/J - y-axis, O/L - z-axis). Use "1" and "2" to switch between unit 1 and unit 2.
    3. Load the primary calibration by clicking on "Load calibration." Click the "Secondary calibration" button on the main GUI under the unit that is in use. For example, if unit 2 is in use, click the "Secondary calibration" button in the "Unit 2" column. Follow the pop-up window instructions to switch to the high-magnification lens.

6. Patch a target cell.
    1. Make sure that the "MultiClamp" (*i.e.* amplifier) software is running. Click on the "Patch control" button to open the "Patch control" GUI; it may take up to a few min to open this GUI.
    2. Use the 40X magnification lens by checking "40X" on the main GUI "Unit 0" column. Click the "go to" button next to the cell of interest on the coordinate list in the "Memory position" GUI; the microscope will move to the cell.
    3. Click on the CTM button of the unit in use in the main GUI to enable movement following a mouse click. Click on the cell of interest; the pipette tip will move to the cell.
    4. Click on the "Patch" button on the "Patch control" GUI.
    NOTE: Automatic patching will begin, and the pressure and resistance can be monitored on the "Patch control" GUI.
        1. Use the "Unit 1 selected" button to switch the input signal between the two units.
        NOTE: The system will approach the target cell, apply negative pressure, match the cell membrane potential, and detect gigaseal formation based on a series of pressure and resistance thresholds and logic.
        2. Manipulate the automatic process at any point by clicking on the respective buttons on the "Patch control" GUI. For example, click on the "Patch" button again to cancel the patch trial and click on "Next stage" to advance the patching process to the next step, regardless of the threshold.
        NOTE: A pop-up window will notify the user when a gigaseal has formed, and the option to apply zap along with large negative pressure will be presented.

    5. Select "Yes" to break in with combined zap and suction. Alternatively, select "No" to break in with suction only.
    NOTE: When a successful whole-cell patch is completed, a pop-up window will remind the user to save the experiment patch log.
    6. Save the experiment patch log.

NOTE: If a patching trial is unsuccessful, a pop-up window will notify the user, and the patch process will reset.
7. Go back to step 2.4 and repeat the steps with a different cell.

7. Refine the patching thresholds (optional).
NOTE: Thresholds for the initial pipette resistance range, positive and negative pressure, gigaseal resistance, *etc*. can be modified from a configuration file.
    1. Open the "PatchControlConfiguration.csv" file in the "Configuration" folder at the destination where the system is installed. Change the numbers corresponding to each threshold value. Do not change the names of the values; this will result in unrecognizable entries in the system.
    2. Implement the new threshold values immediately by pressing "Ctrl+L" without restarting the program; restarting the program will implement the newest threshold value from the file.

3. Performing Recordings

NOTE: The mode in the computer-controlled microelectrode amplifier will be set automatically to Current Clamp ("IC") by the autopatcher software once a successful patch has been achieved. Whole-cell patch clamp recordings can be done using the recording software of choice (this system does not include a recording function). If multiple target cells were identified, after finishing a recording, go back to step 2.4 and try another cell.

1. Perform automatic local drug application experiments using a picospritzer (optional).
NOTE: Here, a local drug application experiment is used as an example to describe how to use the additional "Command Sequence" function to control external hardware through TTL signals.
    1. Connect port A channel 0 and the ground on the secondary DAQ board to the start trigger input on the digitizer using a stripped BNC cable (as described in step 1.2.3). Connect one digital output channel on the digitizer to the external trigger on the picospritzer.
    2. Prepare the picospritzer according to the user manual. Connect the picospritzer air output to the drug-puff pipette holder attached to a micromanipulator.
    3. Load a pipette filled with a drug of choice. Attach it to the pipette holder. Calibrate the pipette, as described in step 1.7.
    4. After patching a cell as described in step 2.6, select a desired location for drug delivery by mouse-clicking in the camera view GUI under "mouse mode" (switched from the main GUI). Alternatively, use the "Grid" GUI to design a grid, with each pixel in the grid as one of the target locations.
    NOTE: The grid can be manipulated in the camera view GUI by dragging with the mouse.
    5. In the "Command Sequence" GUI, select the manipulator unit that the drug pipette is mounted on. Click "load mouse points" or "load grid points" to import all coordinates for drug delivery.
        1. Click on each coordinate entry to edit the specific command TTL signal in the right column. In the first command entry, click on the right-most digit to turn it from "0" to "1," sending a +5-V TTL signal. Set the time (T) to the desired TTL signal duration, in ms.
        2. In the second command entry, set all digits to "0" and set T to the duration of the recording length of the trial. Edit commands for all coordinate entries. Add command entries by clicking "+" if multiple TTL signals are necessary.
        NOTE: The 8-digit bits represent port A channels 0-7 on the secondary DAQ (ports 1 - 3 are occupied by the pump and two valves) can be switched, if necessary.
    6. Create a recording protocol in the data acquisition module so that the start of a sweep is triggered by the external start trigger. Edit the protocol to deliver the drug at the desired time.
    7. In the "Command Sequence" GUI, click "Run" on the left to run all coordinates. Alternatively, click "Run" on the right to run only the selected sequence.
    NOTE: The pipette will move to each coordinate and execute the TTL signal as defined to start the recording sweep.

Representative Results

Our system has been tested on its ability to patch cells in acute brain slices, mouse induced Pluripotent Stem Cells (iPSCs) differentiated into neurons, and HEK 293 cells artificially expressing channels of interest. **Figure 3** shows an experiment using Thy1-ChR2-YFP transgenic mice (B6.Cg-Tg(Thy1-COP4/EYFP)18Gfng/J) targeting fluorescently labeled layer 5 pyramidal neurons in the visual cortex. The target cell was one of the automatically identified green fluorescent-positive cells (**Figure 3b**). **Figure 3a** is the Differential Interference Contrast (DIC) image of the patched neuron. The whole-cell configuration was achieved by the automatic patching protocol in steps 2.5 - 2.6 and was validated by step current injection-induced action potentials (**Figure 3c**)

To demonstrate the additional "Command Sequence" function, we delivered 500 mM KCl for 200 ms to three locations on a brain slice while patching a cell (**Figure 4**). First, we selected 3 locations on the brain slice: one close to the patched cell body and two far away from the patched cell. The coordinates were stored in the "Memory Positions" GUI. The coordinates were loaded to the "Command Sequence" GUI under "Unit1," which was the manipulator that the KCl-containing pipette was mounted on. We set the commands in the left column to send a +5-V TTL signal for 500 ms, followed by 0 V for 10 s (**Figure 4a**), from port A channel 0 on the secondary DAQ board, which was connected to the digitizer "start trigger" input. **Figure 4c** shows that the patched cell was a regular spiking neuron. The drug application pipette (Unit 1) traversed the three selected locations automatically (**Figure 4b**), and we recorded 10 s for each application under voltage-clamp (**Figure 4d**). The color of the traces in **Figure 4d** corresponds to the border color in **Figure 4b**. When KCl was puffed at the cell, a large inward current was observed, which slowly diminished as KCl diffused. Red fluorescent dye was added to the KCl solution to indicate the spatial distribution of drug delivery and was imaged using combined DIC and epifluorescent imaging. This experiment illustrated the ease and flexibility of our system to control manipulator/microscope movement and external hardware through TTL signals.

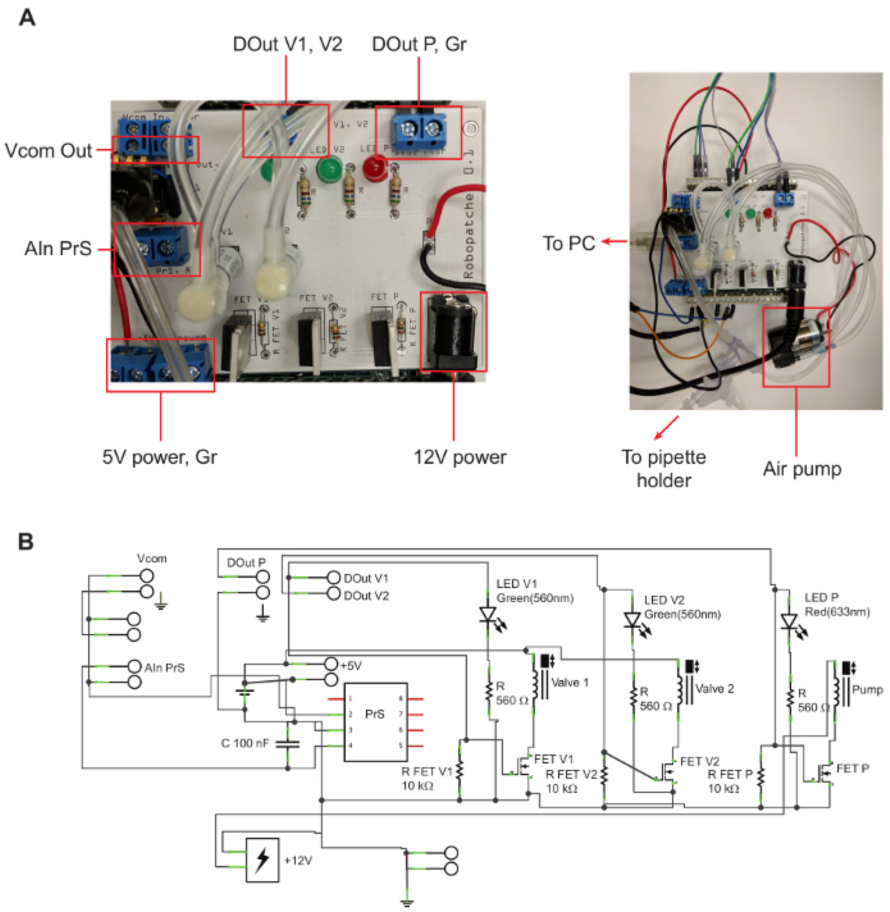


Figure 1. Pressure Control Unit. A: Printed Circuit Board (PCB) for connecting the valves, pressure sensor, and air pump. The left shows details on the PCB, labeling locations of outputs that are mentioned in the protocol. The right shows the connection between the PCB and the air pump, USB port, and tubing. **B:** Circuit map for the PCB. Please click here to view a larger version of this figure.

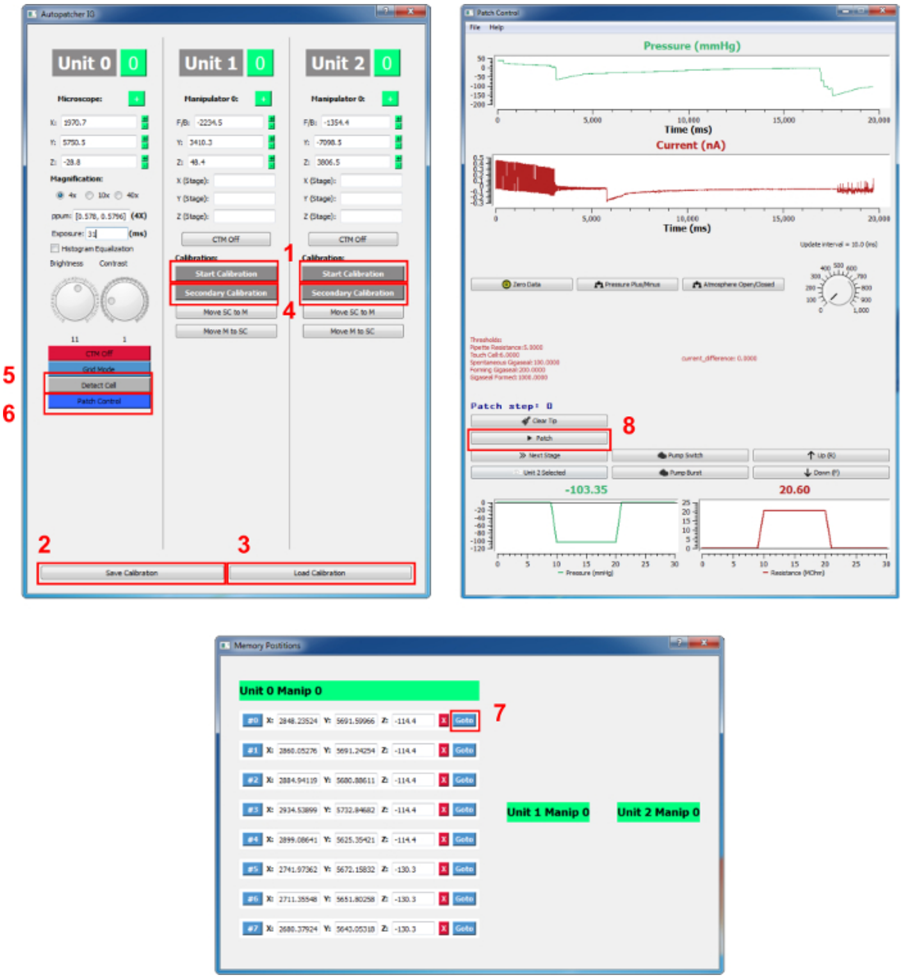


Figure 2. Autopatcher GUI. The buttons mentioned in the protocol are shown in red squares and are numbered. 1: Start Calibration, 2: Save Calibration, 3: Load Calibration, 4: Secondary Calibration, 5: Detect Cell, 6: Patch Control, 7: Go to (target cell coordinate), and 8: Patch. Please click here to view a larger version of this figure.

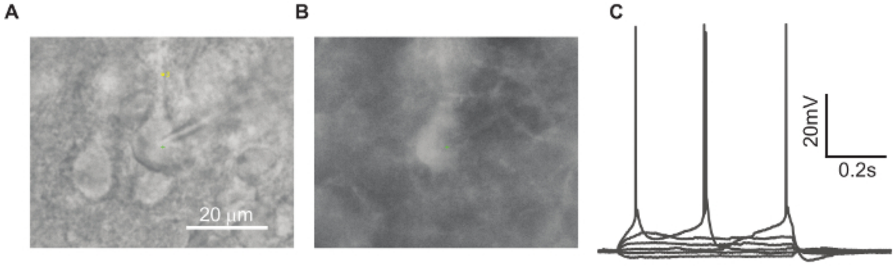
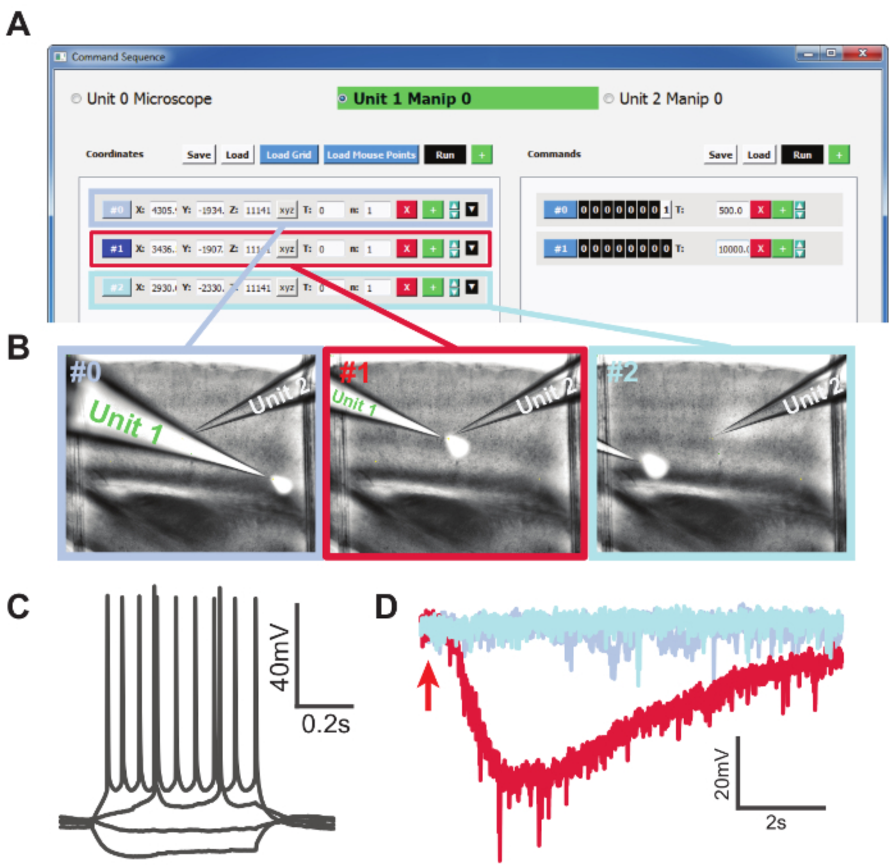


Figure 3. An Example of the Patched ChR2-YFP-positive Cell. A: 40X magnification under DIC optics. **B:** Epifluorescence image of the same cell in panel A (LED illumination at 488 nm). **C:** Current-clamp recordings from the patched cell during a series of hyperpolarizing and depolarizing step current injections. Please click here to view a larger version of this figure.



Command Sequence

○ Unit 0 Microscope ● Unit 1 Manip 0 ○ Unit 2 Manip 0

Coordinates

#0 X: 4305.! Y: -1934. Z: 11141 xyz T: 0 n: 1

#1 X: 3436.: Y: -1907. Z: 111·1 xyz T: 0 n: 1

#2 X: 2930.(Y: -2330. Z: 11141 xyz T: 0 n: 1

Commands

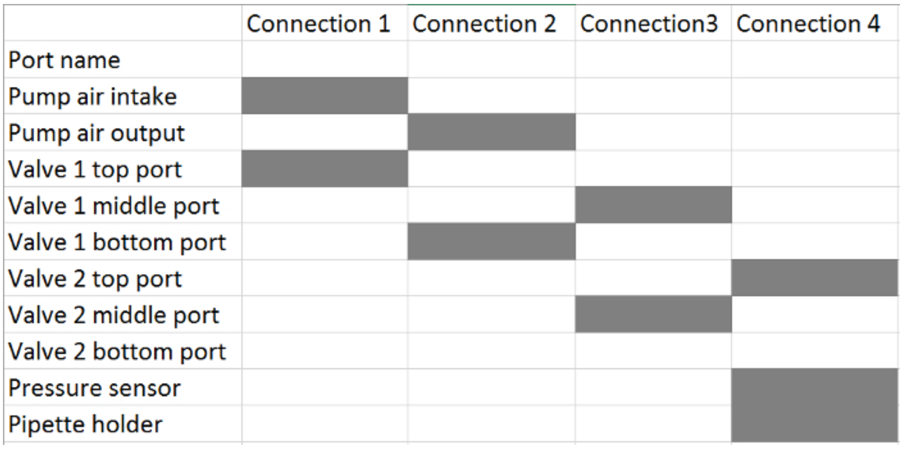
#0 0 0 0 0 0 0 0 1 T: 500.0

#1 0 0 0 0 0 0 0 0 T: 10000.(

Figure 4. Conducting an Automated Drug Delivery Experiment. A: Selected locations loaded to the "Command Sequence" GUI. The left column shows the list of coordinates, and the right column shows the list of commands in the form of TTL signals for each location. **B:** Screenshots during the drug application experiment corresponding to the three selected locations. Unit 1 was the KCl-containing pipette and Unit 2 was the patching pipette. KCl solution was mixed with red fluorescent dye for the purpose of visualization. Images were obtained by combining DIC and fluorescence imaging. **C:** Step current injections showing a regular spiking neuron. **D:** Voltage-clamp recording traces from the local application of 500 mM KCl solution at three locations. The red trace with inward current was recorded from the trial when KCl application was close to the patched cell. The red arrow indicates the timing of KCl application. Please click here to view a larger version of this figure.

Outlet on the PCB	Port name on the DAQ board	Port # on DAQ board	Remark
DOUT V1	Port A channel 1	22	Control valve 1
DOUT V2	Port A channel 2	23	Control valve 2
DOUT P	Port A channel 3	24	Control air pump
Gr	Ground	29	Ground

Table 1. Printed Circuit Board (PCB) to secondary data acquisition (DAQ) board connection configuration. Use this table to connect PCB outputs (first column from left) to ports on the DAQ board (second column from left). The port name and number on the secondary DAQ refer to single-ended mode.



	Connection 1	Connection 2	Connection3	Connection 4
Port name				
Pump air intake	▨			
Pump air output		▨		
Valve 1 top port	▨			
Valve 1 middle port			▨	
Valve 1 bottom port		▨		
Valve 2 top port				▨
Valve 2 middle port			▨	
Valve 2 bottom port				
Pressure sensor				▨
Pipette holder				▨

Table 2. Tubing Connections from the Pressure Control Unit to the Pipette Holder(s). For each connection, connect the corresponding ports, highlighted with a grey box, using soft tubing (see the **Table of Materials**).

Discussion

Here, we describe a method for automatic image-guided patch clamp recordings *in vitro*. The key steps in this process are summarized as follows. First, computer vision is used to automatically recognize the pipette tip using a series of images acquired via a microscope. This information is then used to calculate the coordinate transformation function between the microscope and the manipulator coordinate systems. Computer vision is used to automatically detect fluorescently labeled cells and to identify their coordinates. These steps are integrated with pipette targeting and the automatic patching algorithm using the open-source Python programming language, PyQT, and OpenCV libraries.

Compared to existing *in vitro* patch clamp methods, this system makes significant improvements in the several areas. It minimizes human intervention. This system automates most of the steps in the patch clamp experiment, minimizing the requirement for human intervention. Some of the remaining manual steps, including switching between the low-/high-magnification microscope lenses, can be automated using additional motorized hardware.

The patch-clamp method improves throughput. Patch-clamp experiments using this system achieved higher success rates and shorter times for each trial, contributing to a significant increase in overall throughput. The computer vision algorithm for fluorescent cell detection and pipette tip detection is very robust, and the error rate was very low. The average error for pipette tip detection was 1.6 µm, and the false-positive rate for fluorescent cell detection was 4.9% ±2.25%. A detailed comparison between traditional manual patching and automatic patching has been made[6].

Detailed documentation of experiments is possible. Patch logs of each trial can be saved and analyzed *post hoc*. Such detailed documentation was not previously available for manual patching. This allows for the systematic analysis of patching experiments in unique experimental conditions, cell types, species, and slice preparations.

This method shows compatibility with standard *in vitro* patch clamp equipment. Our system, as demonstrated in this manuscript, is designed to augment existing *in vitro* patch clamp rigs, giving them the capacity to conduct automatic patching. Unlike the planar patch approach, this system is suitable for laboratories already conducting manual patch clamping to convert their equipment at minimal cost. At the same time, there is still the option to patch manually or semi-automatically using the same system.

Because of the adaptability of the system mentioned above, connecting the hardware and configuring the software is required by the experimenter when the system is set up for the first time. Problems may result from incorrect port assignment and inadequate driver libraries for the control of certain hardware. Please refer to steps 1.2 - 1.4 when troubleshooting.

Compared to the partial automation of existing systems, this system achieves the maximum level of automation in the conventional *in vitro* patch clamping of acute brain slices (and other *in vitro* preparations). This is true for all steps, from cell detection to pipette calibration to patching[7,11]. The only bottleneck is the manual process of filling and changing the patch pipettes between trails. Recent developments in the reuse of patch pipettes can potentially solve this problem[12]. Besides the quality of slice preparation, the most common reason for unsuccessful trials originates from manipulator mechanical errors and the movement of the slice in the chamber. These limitations are beyond our control in the current system. Efforts are being made to implement close-loop, real-time detection and control of pipette movement to account for this problem.

For future development, we are interested in expanding the current fluorescent cell detection capabilities to general cell detection under DIC optics.

Disclosures

A non-provisional patent application "SYSTEMS AND METHODS FOR AUTOMATED IMAGE-GUIDED PATCH-CLAMP ELECTROPHYSIOLOGY IN VITRO," U.S. Serial No.: 15/353,719, was filed on November 16, 2016, Ref. No.: PRF 67270-02.

Acknowledgements

We are grateful for the financial support from the Whitehall Foundation. We would like to thank Samuel T. Kissinger for the valuable comments.

References

1. Sakmann, B., Neher, E. Patch clamp techniques for studying ionic channels in excitable membranes. *Annu Rev Physiol.* **46**, 455-472 (1984).
2. Collins, M. D., Gordon, S. E. Giant liposome preparation for imaging and patch-clamp electrophysiology. *J Vis Exp.* **76** (2013).
3. Kodandaramaiah, S. B., Franzesi, G. T., Chow, B. Y., Boyden, E. S., Forest, C. R. Automated whole-cell patch-clamp electrophysiology of neurons in vivo. *Nat Methods.* **9** (6), 585-587 (2012).
4. Desai, N. S., Siegel, J. J., Taylor, W., Chitwood, R. A., Johnston, D. MATLAB-based automated patch-clamp system for awake behaving mice. *J Neurophysiol.* **114** (2), 1331-1345 (2015).
5. Kodandaramaiah, S. B. *et al.* Assembly and operation of the autopatcher for automated intracellular neural recording in vivo. *Nat Protocols.* **11** (4), 634-654 (2016).
6. Wu, Q. *et al.* Integration of autopatching with automated pipette and cell detection in vitro. *J Neurophysiol.* **116** (4), 1564-1578 (2016).
7. Perin, R., Markram, H. A computer-assisted multi-electrode patch-clamp system. *J Vis Exp.* **80**, e50630 (2013).
8. Fertig, N., Blick, R. H., Behrends, J. C. Whole cell patch clamp recording performed on a planar glass chip. *Biophys J.* **82** (6), 3056-3062 (2002).
9. Brown, A. L., Johnson, B. E., Goodman, M. B. Making patch-pipettes and sharp electrodes with a programmable puller. *J Vis Exp.* **20** (2008).
10. Segev, A., Garcia-Oscos, F., & Kourrich, S. Whole-cell Patch-clamp Recordings in Brain Slices. *J Vis Exp.* **112** (2016).
11. Campagnola, L., Kratz, M. B., Manis, P. B. ACQ4: an open-source software platform for data acquisition and analysis in neurophysiology research. *Front Neuroinform.* **8** (3) (2014).
12. Kolb, I. *et al.* Cleaning patch-clamp pipettes for immediate reuse. *Sci Rep.* **6** (2016).