Instruction of Skuli code

Bocheng Yin

July 29, 2019

#### Warning

#### Directly use our Sikulix codes on a computer other than ours will fail with no doubt for many reasons.

* + Different softwares and workflow are applied
  + the Graphical user interface can change because of the variations in operating system, software version, layout and configuration.
  + different scripts are used.

#### Please customize your own sikulix code based on your purpose of automating of multiplex tasks.

#### Please adjust the similarity or replace the icons with the ones freshly captured from your local desktop where the Sikulix code will run, regardless that you will follow the exact same workflow within the same working environment as us.

#### Refers to <https://launchpad.net/sikuli> for help

#### the instruction aims to interpret our workflow and how it is implemented with the code.

#### the quality of the code is limited by my inadequate knowledge and experience in coding.

### source codes

#### Sikuli code 1.“stp\_call\_oneChannel.sikuli”

#### Sikuli code 2.“stp\_becalled\_oneChannel.sikuli”

#### The sikuli code is bundled in a folder with a python script and all the pictures that are cited in the python script.

#### please open the sikuli codes in sikuli Integrated development environment (IDE)

#### download Sikuli IDE from <http://sikulix.com/>

#### the version in which the code is generated in Sikuli IDE version 1.1.4.

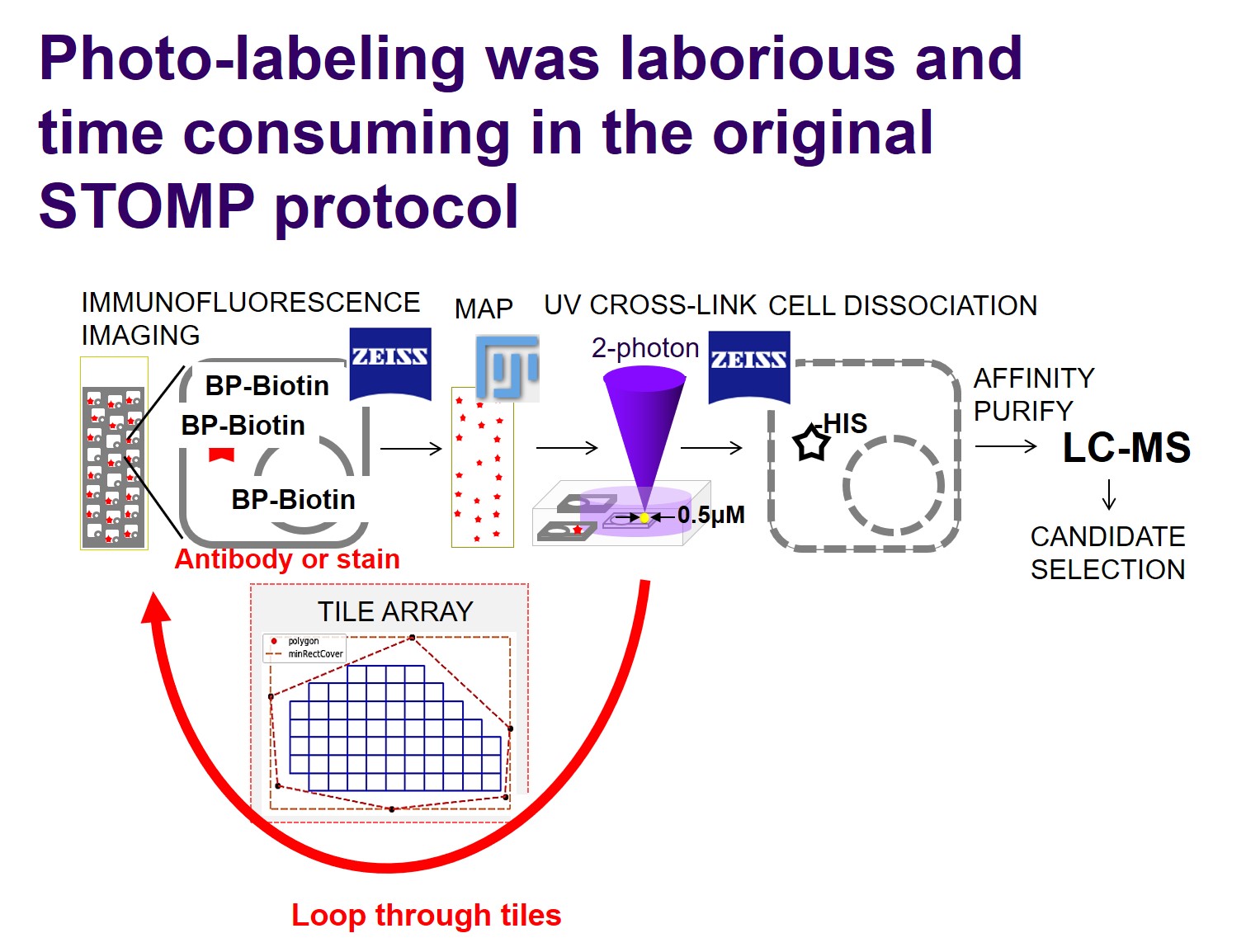
#### Windows version: Windows 7 professional

#### View the sikuli codes

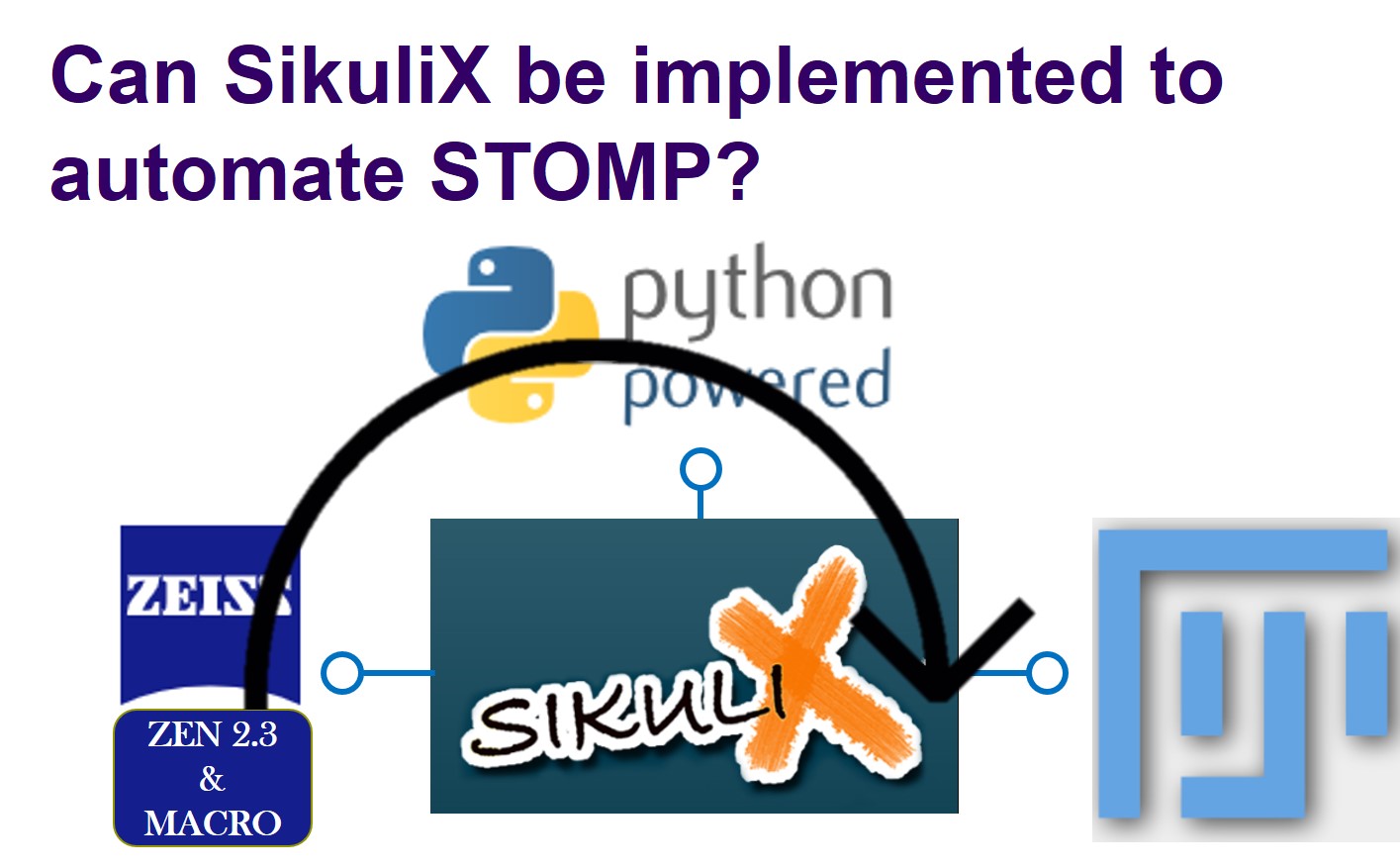
* the codes can be viewed directly in the sikuli IDE
* If your external display screen has a distinct display parameters (i.e. resolution and size) from ours, the codes may not be opened.
* In case you have troubles to view our code in sikuli IDE, HTML versions are posted for read-only, not for execution.
  + stp\_call\_oneChannel.html
  + stp\_becalled\_oneChannel.html

### The task of the sikulix code

#### we need automating the STOMP process

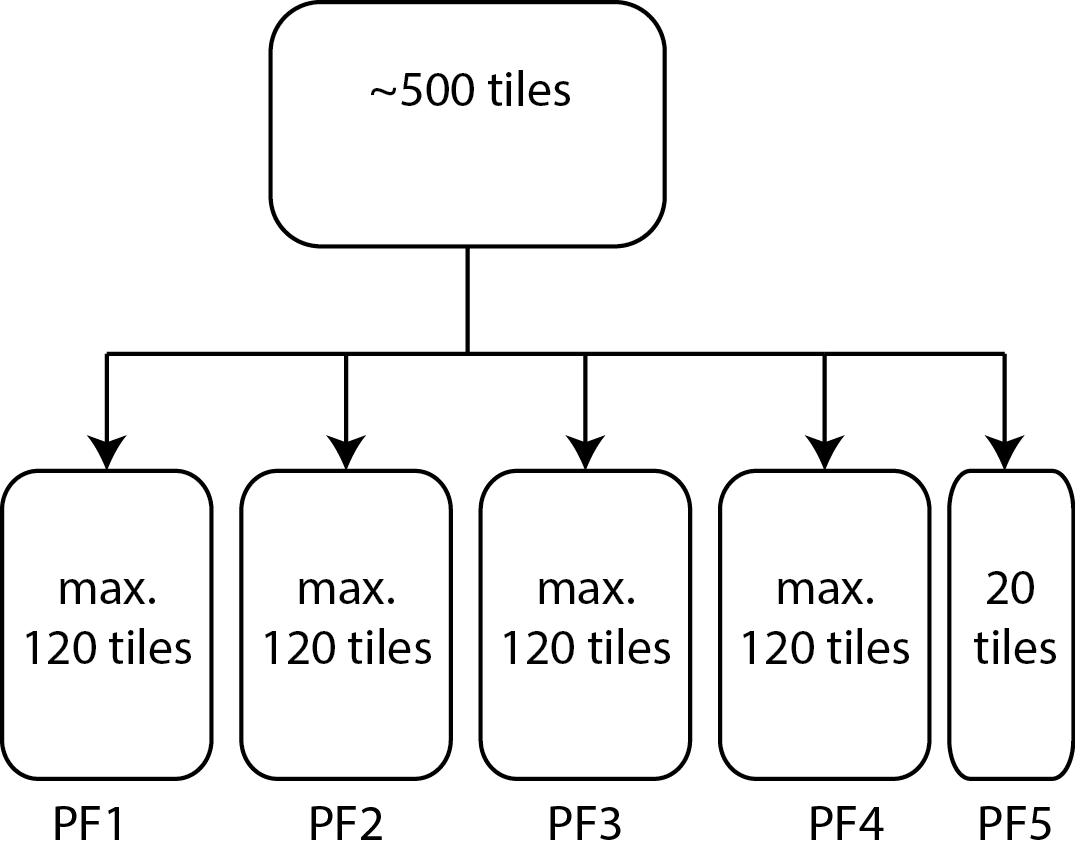
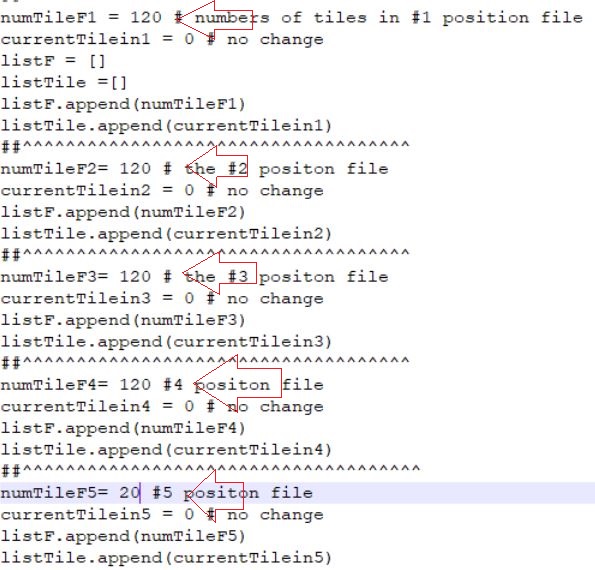
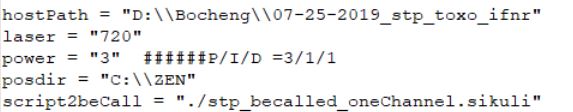


### Use sikulix code to accomplish the automation



### Input parameters for code 1, “stp\_call\_oneChannel.sikuli”

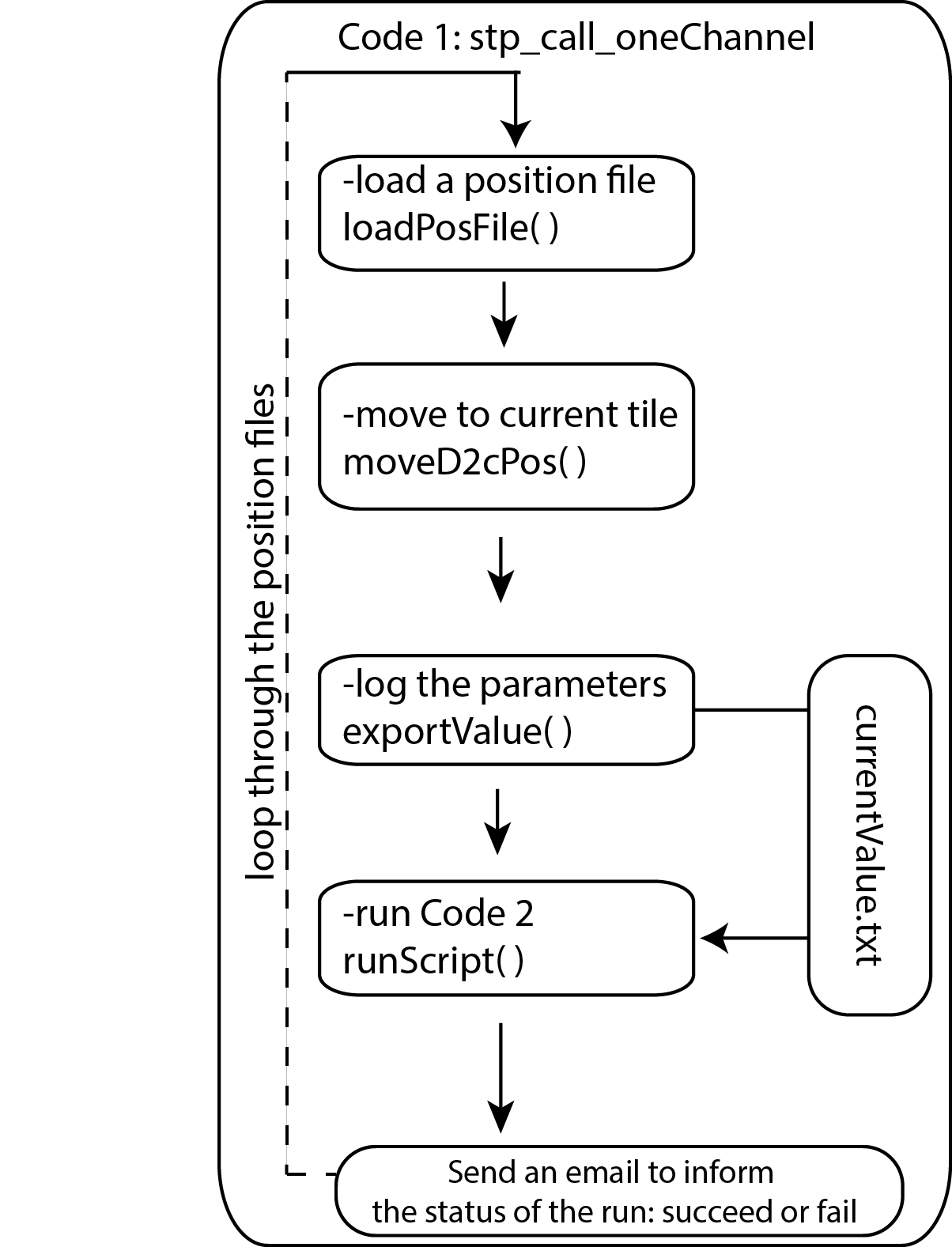
#### the parameters must be entered everytime for a new run

* + because ZEN Black cannot work fluently with more than 120 file positions, we have to split all the tile positions into several position files with maximum 120 tile positions
  + each time only maxium 120 tile positions are loaded into ZEN Black. 
* STOMP starts from position file #1 and tile position #1
* 
* STOMP starts from position file #2 and tile position #51
  + resume the work after the position file#2, tile position #50 where the code crashes.
* 
* enter the tiles contained in each position file here 
* STOMP macro in ZEN Black
  + hostPath: the directory where the images will be stored
    - the folder must be created for every new sample
  + laser parameters:
    - laser: the wavelength of two-photon
    - power: the power of two-photon
  + posdir: the fixed directory to store position files and parameter files
  + script2beCall: the script will be called by “stp\_call\_oneChannel.sikuli” 

### Input parameters for code 2, “stp\_becalled\_oneChannel.sikuli”

* input parameters will be logged into the currentValue.txt by code 1, “stp\_call\_oneChannel.sikuli”
* 

### The directed network of functions in Code 1, “stp\_call\_oneChannel.sikuli”



### The directed network of functions in Code 2, “stp\_becalled\_oneChannel.sikuli”

