ZeissTrial Async Software

Tianbi Zhang

December 2023

Contents

1 Introduction

1.1 Preface

ZeissTrial is a software monitoring and controlling a Zeiss scanning electron microscope (SEM). It was originally developed by Tianbi Zhang for his PhD work on a custom transmission Kikuchi diffraction (TKD)-in-SEM and electron backscatter diffraction (EBSD) setup, where the software facilitates semi-automated beam scan and pattern capture. This project was directly adapted from the demo project shipped along with the SmartSEM programming interface.

In this repository the Async "flavour" is included. There are a few other "flavours" tested internally at the University of British Columbia for pure testing purposes. This software package has a unique digital object identifier (DOI): `https://zenodo.org/doi/10.5281/zenodo.10161482`. Please acknowledge this work when appropriate.

1.2 Glossary

- ZeissTrial: this software.
- Client and server: when you establish a remote connection to the SEM PC, the client is the PC where you run ZeissTrial, and the server is the SEM PC where SmartSEM is installed and being controlled by ZeissTrial.

1.3 System requirements

This project only runs on Windows operating systems aka Windows 7–10. **It does not work on Windows 11 due to references to 32-bit libraries!** Other requirements include:

- .NET Framework version 4.0 or higher for C# and an integrated development enviroment such as Visual Studio. To avoid .NET version issues, VS 2019 is suggested.
- The Zeiss SmartSEM API (proprietary) is not included in this repository. You need to add the reference of the APILib manually.
- The user should check that all the inclusions and packages used in the C# projects are imported.

- You need access to Zeiss SmartSEM and setup the remote connection, even if you want to control an emulator or control SmartSEM on the same PC as the microscope control PC. This requires a remote connection.
  - In general, you will need to set up the configuration in RConfigure.
  - If you run ZeissTrial on the same PC as SmartSEM, you can just use localhost.
  - If run remotely, you will need to connect the ZeissTrial PC to the same network as the SmartSEM PC and assign a static IP to the ZeissTrial PC. You can simply test it by trying to establish a remote connection (Initialize Remote).

2 Overview of Zeisstrial Async

For convenience, the GUI window and all components are divided into a few zones (numbered 1-6). An overview of each zone and the associated components is shown below.

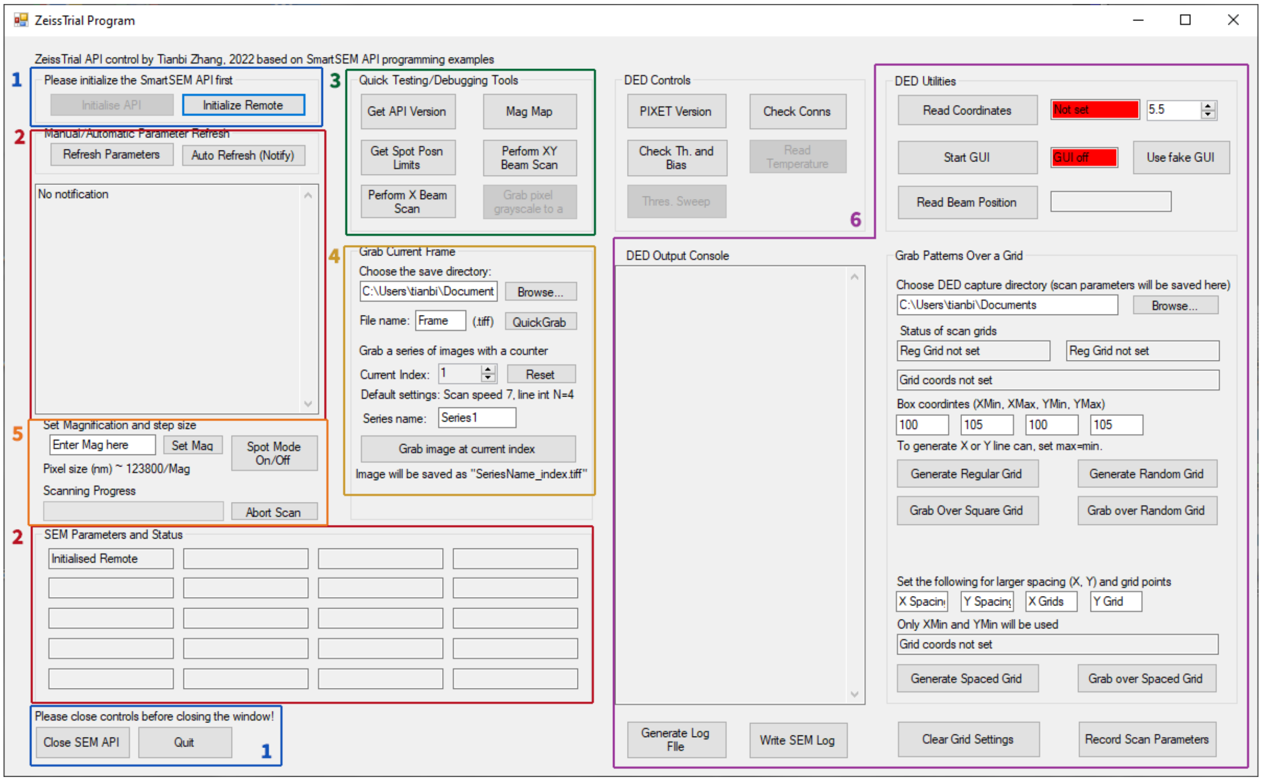

Figure 1: The GUI of ZeissTrial Async

2.1 Initialization and Disconnection Zones

- “Initialize API”. This button establishes a remote connection with SmartSEM when SmartSEM runs locally on the same PC. Note that “localhost” must be set in RConfigure first.
- “initialize Remote”. This button establishes a remote connection with SmartSEM on another PC. Note that the appropriate IP addresses must be set in RConfigure first.
- “Close SEM API”. This button closes the SmartSEM API.
- “Quit”. This button exits the program.

If you initialize the local API, the ”Initialize Remote” button will gray out. A message box will pop up and indicate whether the API has been correctly initialized, and the first parameter in the SEM parameter and status zone will show ”Initialized” instead of ”Not Initialized”. You must initialize the SEM API before calling any SEM functions; otherwise, an error box will show up. Before exiting the program, the SEM API must be closed.

In order to establish the remote connection, a SmartSEM user must log in on the client first.

2.2 Parameter Monitoring Zone

- “Refresh Parameters”. This reads a selected number of SEM parameters and refreshes the SEM parameters and status zone.
- “Auto Refresh (Notify)”. When enables, SEM parameters abd status zone will be automatically refreshed when there is a change to their value/status.
- SEM parameters and status zone. This is where a selected number of SEM parameters and status are displayed. By default, upon starting the GUI, no parameters are displayed.
- Notifications zone. This is a textbox to show SEM API related outputs, including error messages.

The parameters and status zone are set up in the following pattern:

You may choose to modify what is displayed. In future versions, “Detector Channel” will be replaced by chamber pressure.

API Initialization State	Actual Current	Magnification	Detector
Vacuum State	Extractor Current	Pixel Size	Detector Channel
Beam State	Spot Mode State	Working Distance	Brightness
EHT Voltage	Scan Speed	Aperture Size	Contrast
High Current Mode State	Stage X Position	Stage Y Position	Stage Z Position

2.3 Quick Testing Zone

ZeissTrial is build on some key API functions in the programming interface: Get, Set, GetLimits, Excecute and Grab. A few quick testing tools are programmed to test if these functions are working properly.

- “Get API Version”. This will simply test if the API is responding. A message box will pop up with the version code.

- “Get Spot Position (Posn) Limits”. This tests if GetLimits works properly and will return with the frame size in a pop-up message box.

- “Perform X Beam Scan”. This tests the Get and Set functions. It will first change to spot mode if it is not on, then read the current position of the spot using “Get” and move the spot horizontally across the entire frame using ”Set”. A message box will appear when the scan is complete. This process can be canceled by clicking the ”Abort” button. Spot mode should be switched on for this mode.

- “Mag Map”. This function will set the magnification to the minimum (using “GetLimits”) in increments of 20 x until 100 kx, read the pixel size (in m) simultaneously, and generate a text file ”magmapvalues.txt” in the selected directory. This function is useful for establishing the relationship between magnification and pixel size (step size), since we cannot set directly the pixel size (it is set through setting the magnification). This process can be canceled by clicking the “Abort” button, and the text file will still be generated. *This button will be disabled in future releases.*

- “Perform XY Beam Scan”. This is similar to ”Perform X Beam Scan” and can be seem as multiple X beam scan in series. The beam will scan across the entire frame in a raster fashion: left to right, then top to bottom. A message box will appear when the scan is complete. This process can be canceled by clicking the ”Abort” button. Note that due to the beam dwell time, it may take a long time for this function to run to completion. *This button will be disabled in future releases.*

X beam scan, Mag Map and XY Beam scan can be interrupted by clicking “Abort Scan” in Zone 5.

2.4 Image Capture Zone

A few functions are provided to capture images.

- Setting capture directory. Images captured will be stored in this directory. Make sure you have write permissions or an error will show up. By default this is set to “My Documents”. You can change this using the “Browse...” window.

- “QuickGrab” button. This captures the current scene in the SmartSEM window. By default, images are captured with “line integral” noise reduction (CMD_LINE_INT) with $N = 4$ and scan speed is set to 9. You can change these parameters in your own version. The function will then get the estimated scan time and wait till the scan is completed and then save the image. The image will be saves as a .tiff image. You can provide a name at the text box at the “File name:” prompt.

- Grab image in a sequence. This can be helpful if you capture a series of images in an automated or semi-automated routine. Similar to the “Quickgrab” button, clicking the “Grab image at current index” button will initiate a scan at scan speed 7 and line integral with $N = 4$ (again, these can be edited in the code). The image captured will be names “SeriesName_index.tiff”, where SeriesName is given in the textbox above (Series1 by default) and index given in the “current index” numeric up-down box (1 by default and can be reset using the “reset” button).

Note that in this software there is no accompanying code to the “grab image in series” mode. The user may want to develop a flavour of their own.

2.5 Scanning (4D-STEM) Zone

Generally speaking, in the 4D-STEM mode, the electron beam is moved to a given coordinate (X, Y) where X, Y are positive integers, and a diffraction pattern is captured by an external detector. Then the electron beam is moved to another coordinate. ZeissTrial Async here acts as a scan engine.

Theoretically, this should work with any detector API with C# support. However, due to the author's limited programming proficiency, Zeisstrial Async simply performs a mouse click at a button on a Python GUI controlling the detector. This is used in our static EBSD work here. You can find the Python GUI here.

To perform a scan, the user needs to set up the scan grid, the mouse click position and time, and start the Python GUI.

2.5.1 Set up the scan grid

There are three types of grids that can setup.

- "Regular grid": the beam moves pixel by pixel in a conventional, zigzag fashion (scan through X first, move on to the next Y, and so on). Step size is 1 pixel. The user needs to provide starting and ending X and Y coordinates.
- "Random grid": similar to a regular grid, but the scan sequence is (pseudo-)randomized using `OrderBy(a => rng.Next())`. Step size is 1 pixel. The user needs to provide starting and ending X and Y coordinates.
- "Regular spaced grid": similar to a regular grid, but step size in X and Y are user inputs and can be multiple pixels (but cannot involve fraction of a pixel). The user needs to provide starting X and Y coordinates, spacing in X and Y in pixels, and number of grid points in X and Y.

After putting in the required parameters, the user need to generate the scan grid first by clicking the corresponding "Generate xxx Grid" button first. The status boxes will indicate if the grid has been set. To help select a region of interest, the "Read Beam Position" button can be used to read the current position of the beam (spot mode must be on in SmartSEM), and the position will show up in the box to thr right.

You can only set up one scan grid at a time. Newer grids settings will overwrite the previous one.

2.5.2 Preparing for external detector control

As is mentioned, the current ZeissTrial Async works along with the Python GUI for the detector. Thus, the Python GUI must be running by clicking the "Start GUI" button to start it. This is achieved by calling the Python interpreter to run the corresponding GUI code. You need to set this up manually in the code. If the GUI successfully starts, the status indicate will turn green and says “GUI on”. For debugging purposed, you can use the “Use fake GUI” to pretend that the GUI is open.

You also need to set up a position for the automated mouse click. Click “Read Coordinates” and hover the cursor to the place you want to click. The program will wait for 3 seconds and then read the cursor position.

As ZeissTrialAsync cannot get the output directly from the Python GUI, we need to manually set a waiting time before the beam moves to the next position. This waiting time (in seconds) can be set at the numeric up-down box on the right (5.5 by default).

Then the scan can be initiated by clicking the corresponding “Grab over xxx grid” button. The output console will provide some outputs of the point being scanned, which can be used to generate a log file using the “Generate Log File”.

Current SEM parameters can be recorded to a text file using the “Record Scan parameters” button. By default, scan type (regular/random/spaced), coordinates and spacing, grid points, magnification, SEM pixel size and SEM stage positions (X, Y and Z) are recorded.

Progress of the scan is indicated by the progress bar in Zone 5. The scans can be interrupted using the “Abort Scan” button in Zone 5.

3 Changelog

- October 2022: initial release (readme.md)
- November 20, 2023: modifications for release.
- December 15, 2023: convert readme.md to the current PDF file.