Preliminary report

Increasing easyLEED's usefulness in research

Hanna Salopaasi
0358645

Table of Contents

Introduction 2
1.1 Background 2
1.2 Objectives and limitations 2
1.3 Structure 3
Literature review 4
2.1 What LEED research is? 4
2.2 HotLEED 4
2.3 Python & Qt 5
Methods of resolution 6
Schedule 7
References 8

Introduction

This is a preliminary report of Bachelor's Thesis done based on summer job in the Department of Mathematics and Physics in Lappeenranta University of Technology. This report's objective is to make a start in increasing the usefulness of the program easyLEED in LEED research.

1.1 Background

EasyLEED is a program designed for the extraction of intensity-energy spectra from low-energy electron diffraction (LEED) patterns. LEED is one of the experimental methods to infer characteristics about the atomistic structure of surfaces [2, 3, 10, 12, 19, 21]. The main reason for easyLEED to excist is that it can calculate the intensity-energy spectra from .fit or .fits types of image files that are used by some of the researchers while others use .img files [15].

EasyLEED is originally created by Andreas Mayer from the Department of Physics, Georg-August-Universität Göttingen in the Department of Physics, Penn State University. The method implemented in the program combines the knowledge about the movement of the spots with detected spot positions using a Kalman filter. [10]

The program is written in Python 2.7 with the graphics done in Qt's Python version PyQt4.

1.2 Objectives and limitations

The main goal of this work is to make easyLEED ready for publishing and getting it out for researchers to use. This means making it more usable in the LEED research process. In practice this happens by atleast making the program multi-platform, adding the possibility to plot the acquired intensity-energy spectra within the easyLEED program, plot the average of the

intensities and the possibility to export the plot(s). Making the user interface (UI) more user-friendly by adding some quick buttons and keeping the menu structure simple and intact are high on the list aswell.

The new code needs to be infused with the old one and be cleaned from excess functions and classes and some comments need to be added to make the code more readable. Documentation requires atleast some sort of quick-start tutorial. The final objective is to make some smart installation for the program.

The work is limited mostly to the UI part of the easyLEED program and it is intended not to interfere with the original and calculational part made by Andreas Mayer. As Mayer still takes part in the programming clashing code must be avoided.

1.3 Structure

Literature review takes a deeper look in the LEED research and there will be a description on some tools used by the researchers and what features in those are desired. There is a short explanation about Python and its libraries used in the easyLEED program.

The ways in which the research value of the program may be increased are discussed in the methods of resolution chapter. There will be descriptions of those Python libraries that may be needed to reach the objectives described in the introduction.

The schedule of the Bachelor's Thesis is set in the schedule chapter.

Literature review

This chapter describes the basics of LEED research and the Python libraries used in the easyLEED program.

2.1 What LEED research is?

In LEED research electrons are accelerated by an electric field in the direction of a sample and are diffracted by the surface of the sample. A fluorescent screen, which is monitored by a digital camera, detects the diffracted electrons [4, 5, 9, 14]. The diffraction maxima are visible on the acquired images as bright spots on a darker background (Fig. 1).

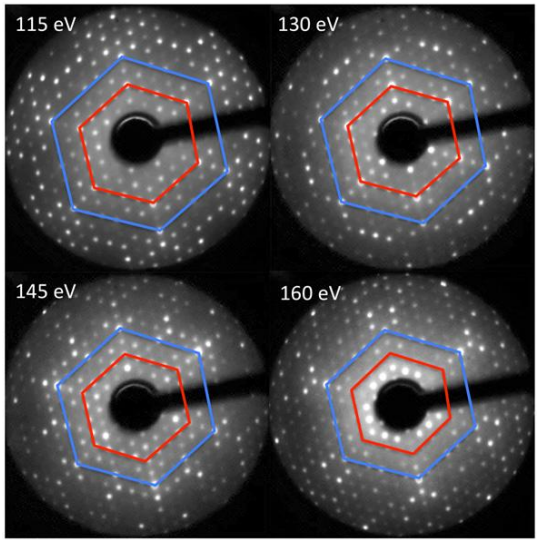

Fig.1. Example of LEED patterns in different energies

In a typical experiment, different accelerating voltages and thus beam energies are used. Information about the position of individual atoms in the surface of the sample can subsequently be obtained from the intensity–energy (I(E)) spectra of the diffraction spots. In order to perform quantitative structural analyses, spectra calculated from theoretical models [6, 11, 20] are fitted to the experimental I(E) spectra. [10]

2.2 HotLEED

HotLEED is a program designed to run on Microsoft Windows™ for the acquisition and the analysis of LEED patterns, both statically and dynamically. The analysis can be performed both in real time and with stored pictures. [8]

One of the main advantages of HotLEED is that it can do the intensity-energy spectras inside the program itself. Saving the spectra is also available as well as the possibility for the user to modify some settings used to calculate the intensities. [8]

2.3 Python & Qt

Python is a high level dynamic, object oriented programming language. It is intuitive and works on all major operating systems and is also under an open source license that makes it freely usable and distributable. Different languages and frameworks are easily integrated to Python which makes it fit for all kinds of tasks. [17]

Qt's PyQt version is one of the frameworks integrated to Python. Qt is a cross-platform application and UI framework primarily developed and maintained by the intrepid developers at Qt Development Frameworks, a unit within Nokia. It is licensed under both open source and commercial license making it usable in open source projects. Qt allows you to write advanced applications and UIs and its high modularity makes easy and quick to use. [13]

Methods of resolution

To achieve the objectives set in the introduction, i.e. multi-platformity, plotting the intensity-energy spectra and making intuitive menu structure, the programming language and main libraries are going to be the same as in the original code. Any additional libraries used are going to be multi-platform and need to be tested on different platforms.

PyQt4 has good menu and toolbar widgets to easy access to functionalities with the possibility to set keyboard shortcuts [1]. So the most natural way to start adding the value of the program is to make use of these widgets and add some quick access buttons in a new toolbar.

Infusing the new code with the old one and cleaning the code from unused functions and classes is best achieved by studying the original code and learning which does which in it. This requires further learning of object oriented Python and the Qt library as well as studying the basics of the methods used in the calculational part. Andreas Mayer has promised to clean up most of the calculational part when he is improving the tracking algorithms but this makes the final program harder to build as all the different versions of the code available.

Improving the documentation by adding a tutorial is one of the easiest parts to do and a simple word editor should suffice in making it. Additional help could be added to the program itself and PyQ4 library needs to be research for a simple method to add it.

There is multiple options for smart installation. In Unix environment a package will probably be enough but for Windows users the program will need to be converted to an .exe format. There are number of programs to do this with, for example PyInstaller and py2exe, and the best option will need to be evaluated by at least the size of the file and the state of Python installation needed in addition to the .exe [16, 18].

Matplotlib is a Matlab like Python library and it can be used for the plotting. [7]

Schedule

4.6.2012	Presentation of the preliminary report and submission the subject request
4.7.2012	At least half of the objects reached and written down
3.8.2012	Most of the work written, suggestions by the instructor
17.8.2012	Final advises and suggested modifications by the instructor before the presentation
24.8.2012	Presentation of the Bachelor's Thesis

References

[1] Bodnar, Jan, *PyQt4 Tutorial*, URL: http://zetcode.com/tutorials/pyqt4/ (31.5.2012)

[2] Gavaza, G. M, Yu, Z.X., Tsang, L., Chan, C.H., Tong, S.Y., Van Hove, M.A., Phys. Rev. B 75 (2007) 014114.

[3] Gavaza, G. M, Yu, Z.X., Tsang, L., Chan, C.H., Tong, S.Y., Van Hove, M.A., Phys. Rev. B 75 (2007) 235403.

[4] Giergiel, J., Hopster, H., Lawrence, J.M., Hemminger, J.C., Kirschner, J., Rev. Sci. Instrum. 66 (6) (1995) 3475–3479.

[5] Heinz, K., Rep. Prog. Phys. 58 (1995) 637.

[6] Held, G., Braun, W., *CLEED manual*, unpublished.

[7] Hunter, John, Dale, Darren, Droettboom, Michael, *matplotlib: python plotting*, URL: http://matplotlib.sourceforge.net/ (31.5.2012)

[8] Leatherman, Gerald S., Ferralis, Nicola, Diehl, Renee D., *HotLEED user manual*, URL: http://xenon.phys.psu.edu/Links_files/HotLEEDusermanual2008.doc (31.5.2012)

[9] Leatherman, G.S., *Low-energy electron diffraction study of alkali metals adsorbed on silver (111), and their coadsorption with rare gases*, PhD thesis, Pennsylvania State University, 1996.

[10] Mayer, Andreas, Salopaasi, Hanna, Pussi, Katariina, Diehl, Renee D., *A novel method for the extraction of intensity–energy spectra from low-energy electron diffraction patterns*, URL: http://www.sciencedirect.com/science/article/pii/S0010465512000811 (31.5.2012)

[11] Moritz, W., J. Phys. C: Solid State Phys. 17 (1984) 353.

[12] Moritz, W., Landskron, J., Deschauer, M., Surf. Sci. 603 (2009) 1306–1314.

[13] Nokia Corporation, *Qt - Cross-platform application and UI framework*, URL: http://qt.nokia.com/ (31.5.2012)

[14] Ogletree, D.F., Blackman, G.S., Hwang, R.Q., Starke, U., Somorjai, G.A., Katz, J.E., Rev. Sci. Instrum. 63 (1992) 104–113.

[15] Pennsylvania State University, *Surface Physics at Penn State*, URL: http://xenon.phys.psu.edu/Links.htm (31.5.2012)

[16] PyInstaller, *PyInstaller*, URL: http://www.pyinstaller.org/ (31.5.2012)

[17] Python Software Foundation, *About Python*, URL: http://www.python.org/about/ (31.5.2012)

[18] Retzlaff, Jimmy, *py2exe*, URL: http://www.py2exe.org/ (31.5.2012)

[19] Soares, E.A., de Castilho, C.M.C., de Carvalho, V.E., J. Phys. Condens. Matter 23 (2011) 303001.

[20] Van Hove, M.A., Moritz, W., Over, H., Rous, P.J., Wander, A., Barbieri, A., Materer, N., Starke,
U., Somorjai, G.A., Surf. Sci. Rep. 19 (3–6) (1993) 191–229.

[21] Watson, P.R., Van Hove, M.A., Hermann, K., NIST Surface Structure Database Version 5.0, NIST Standard Reference Data Program, Gaithersburg, 2003.