Lappeenranta University of Technology
The Faculty of Technology Management
Degree Program in Information Technology

Bachelor's Thesis

Hanna Salopaasi

Developing a non-public research software for public use

Examiner Associate Professor Uolevi Nikula

Abstract

Lappeenranta University of Technology
The Faculty of Technology Management
Degree Program in Information Technology

Hanna Salopaasi

Developing a non-public research software for public use

Bachelor's Thesis

24 pages, 7 figures

Examiner: Associate Professor Uolevi Nikula
Keywords: LEED, low energy electron diffraction, Python, PyQt, software development

EasyLEED is a program designed for the extraction of intensity-energy spectra from low-energy electron diffraction patterns. It can be used to get information about the position of individual atoms on a surface of some substance. The goal of this thesis is to make easyLEED useful in LEED-research. It is achieved by adding new features, i.e. plotting intensity-energy spectra, setting tracking parameters and allowing exporting and importing of settings and spot location data, to the program. The detailed description of these added features and how they're done and how they impact on the usefulness of the program in research are presented in this thesis. Improving the calculational part of the program is not discussed.

Tiivistelmä

Lappeenrannan teknillinen yliopisto
Teknistaloudellinen tiedekunta
Tietotekniikan koulutusohjelma

Hanna Salopaasi

Yksityisen tutkimusohjelman kehitys julkiseksi

Kandidaatintyö

24 sivua, 7 kuvaa

Työn tarkastaja: Tutkijaopettaja Uolevi Nikula

Hakusanat: LEED, matalaenergiaelektronidiffraktio, Python, PyQt, ohjelmistokehitys
Keywords: LEED, low energy electron diffraction, Python, PyQt, software development

EasyLEED ohjelma on suunniteltu laskemaan intensiteetti-energiaspektri matalaenergiaelektronidiffraktiokuvioista. Sitä käytetään saamaan tietoa yksittäisten atomien sijainnista aineen pinnalla. Tämän työn tarkoitus on lisätä easyLEEDin hyödyllisyyttä LEED-tutkimuksessa ja mahdollistaa sen jatkokehitys. Hyödyllisyyden lisääminen saavutetaan lisäämällä ohjelmaan uusia ominaisuuksia, kuten intensiteetti-energiaspektrin kuvaajan tulostus, seurantaparametrien muokkaus ja mahdollisuus tuoda ja viedä seuranta-asetuksia ja pisteiden sijainti dataa. Tässä työssä esitetään yksityiskohtainen kuvaus lisätyistä ominaisuuksista, miten ne ovat tehty ja millainen vaikutus niillä on ohjelman tutkimuskäytettävyyteen. Ohjelman laskennallista puolta ei käsitellä.

Preface

This Thesis is done based on a summer job in the Department of Mathematics and Physics in Lappeenranta University of Technology and I'd like to thank Katariina Pussi for employing me and giving me this opportunity to work on this. I'd also like to thank my co-workers for a great relaxed atmosphere and the litres of coffee consumed together.

A big thanks goes for the examiner Uolevi Nikula for helping me through this work and correcting my mistakes.

Table of Contents

Abbreviations

EPub	Electronic Publication
HTML	Hypertext Markup Language
LEED	Low Energy Electron Diffraction
PDF	Portable Document Format
UI	User Interface

1. Introduction

This is a Bachelor's Thesis done based on a summer job in the Department of Mathematics and Physics in Lappeenranta University of Technology. This thesis is done as a part of the Degree Program in Information Technology and it's objective is to increase the usefulness of the program easyLEED in mathematical, low energy electron diffraction (LEED), research.

1.1 Background

EasyLEED is a program designed for the extraction of intensity-energy spectra from low-energy electron diffraction patterns. LEED is one of the experimental methods to infer characteristics about the atomistic structure of surfaces [3, 4, 12, 13, 20, 22]. Common factor in programs used in the research is that they use file types that include a header in which information such as the energy used can be stored. Where easyLEED alters from other available programs is that it can use different file types to calculate the spectra than the other programs, such as fit or fits while others use img files [16].

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. [11]

The program is written in Python 2.7 with the graphics done in Qt's Python version PyQt4. It consists of around 1500 lines of code divided into four modules: configuration module, graphics and data fetching module, base functionality module which includes for example reading file headers and test suite module which has functionalities in test phase.

1.2 Objectives and limitations

The main goal of this thesis is to make easyLEED more user-friendly so it does not need a specialist to use. In practice this happens by making the program multi-platform, i.e. working on Windows and Linux, 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 requirements as well. In addition the new code needs to be infused with the old one and be cleaned from excess functions and classes and clarifying comments need to be added to make the code more readable. Documentation requires at least some sort of quick-start tutorial. The final requirement is to make a smart installation for the program.

The work is restricted 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, overlapping code must be avoided. Other constraints include the use of PyQt4 and Python 2.7 and that the language of the program and the code is English.

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 section. There will be descriptions of those Python libraries that may be needed to reach the requirements described in the section 1.2.

The resolution section describes what features were were added to the easyLEED program during this work. The usefulness of these features and some alternative methods of resolution will also be discussed.

Finally the success of the work will be evaluated and the work done summarised in the analysis section.

2. 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 [5, 6, 10, 15]. The diffraction maxima are visible on the acquired images as bright spots on a darker background as seen in the figure 1.

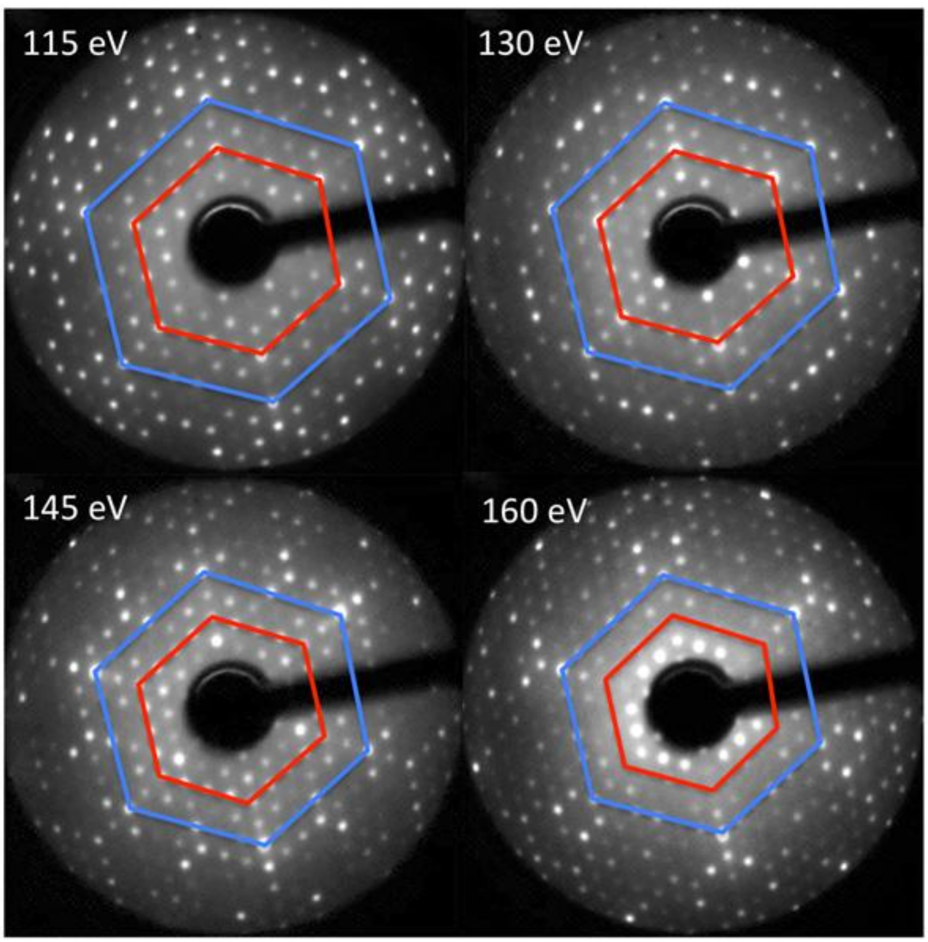

Fig.1. Example of LEED patterns in different energies

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 [7, 12, 21] 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. HotLEED is done by Gerald S. Leatherman, Nicola Ferralis and Renee D. Diehl. [9]

One of the main advantages of HotLEED is that it can do the intensity-energy spectra inside the program itself. Saving the spectra is also available as well as the possibility for the user to modify settings used to calculate the intensities. [9]

HotLEED lacks fit/fits and Linux support and easyLEED is developed to include these.

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 scientific programs. [18]

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 the writing of advanced applications and UIs and its high modularity makes it easy and quick to use. [14]

3. Methods of resolution

Usability improvements in the initial requirements set for the thesis are working on both Windows and Linux, plotting the intensity-energy spectra, making intuitive menu structure, smart installation and tutorial. To make the program work on Windows and Linux the libraries used in the code need to work on both of the platforms. Python and PyQt4 are the main language and framework as restricted so they're multi-platform by default. For plotting the intensity-energy spectra a Python library called matplotlib can be used. It is based on Matlab and has working versions on both of the defined platforms [8]. Menu structure can be maintained and improved by PyQt4 menu and toolbar widgets. Toolbar makes it easier to access mostly used functionalities and keyboard shortcuts can be set for quick access [1]. There is multiple options for smart installation. In Linux environment a package will probably be enough but for Windows users the program will need to be converted to an executable file that possibly includes the required Python libraries. There are number of programs to do this with, for example PyInstaller and py2exe, and the best option will need to be evaluated by the size of the file and the state of Python installation needed in addition to the executable [17, 19]. A simple tutorial and user-guide are easy to make with a normal word editor. An additional help section could be added to the program.

As for maintainability improvements infusing new code with the old one and cleaning the code from unused functions and classes is best achieved by getting to know the original code. This requires further learning of object oriented Python and the PyQt4 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. Improving the documentation by commenting the code will be a major increase in maintainability.

4. Resolution

All the initial requirements, i.e. working on Windows and Linux, adding the possibility to plot the acquired intensity-energy spectra within the easyLEED program, plot the average of the intensities, the possibility to export the plot(s), user-friendly UI, code cleaning and commenting and smart installation, set to the project have been done. Some features that weren't originally planned have also been added to the program.

The following features were added to easyLEED:

- plotting the intensities and the average of the intensities
- widget for selecting plotting method
- quick access toolbar
- exporting the plot as PDF (Portable Document Format) file
- exporting/importing the initial spot locations
- widget for setting tracking parameters

Program has also been tested and it is working on both Windows and Linux. The code has been merged with the original branch that Andreas Mayer is working on. Same naming principles are used throughout the code and all the functions are commented and cleaned from debug lines. The program no longer contains unused lines of code and the structure has been modularised. It has around 1900 lines of code in five modules: functionality, file handling, graphics, default configuration and tracking algorithm. This clear structure makes it easier for other people to develop the program further to accommodate their research needs.

The code of easyLEED follows the form of a Linux package so it is easy to use for a Linux user. For Windows users an installation file was made with PyInstaller. The file contains a Python 2.7 installation and all the necessary libraries needed to run the program. For Linux the library dependencies are listed in the user manual. PyInstaller was chosen because of its ability to make a standard executable that runs without needing a Python installation on the computer. The program aims to be simple to use: just download the executable file and start using it.

4.1 Plotting

Plotting is done with the matplotlib Python library. Originally it was done with the matplotlib's built-in UI but it did not work on Windows because of a faulty library. The plotting raised an "Fatal Python error: PyEval_RestoreThread: NULL tstate" error. After some research on the error the solution was to embed the plotting to a PyQt4 widget which solved the problem as the error is in the part of the library drawing the plot. When embedded to a graphical widget a different function is used to show the plot.

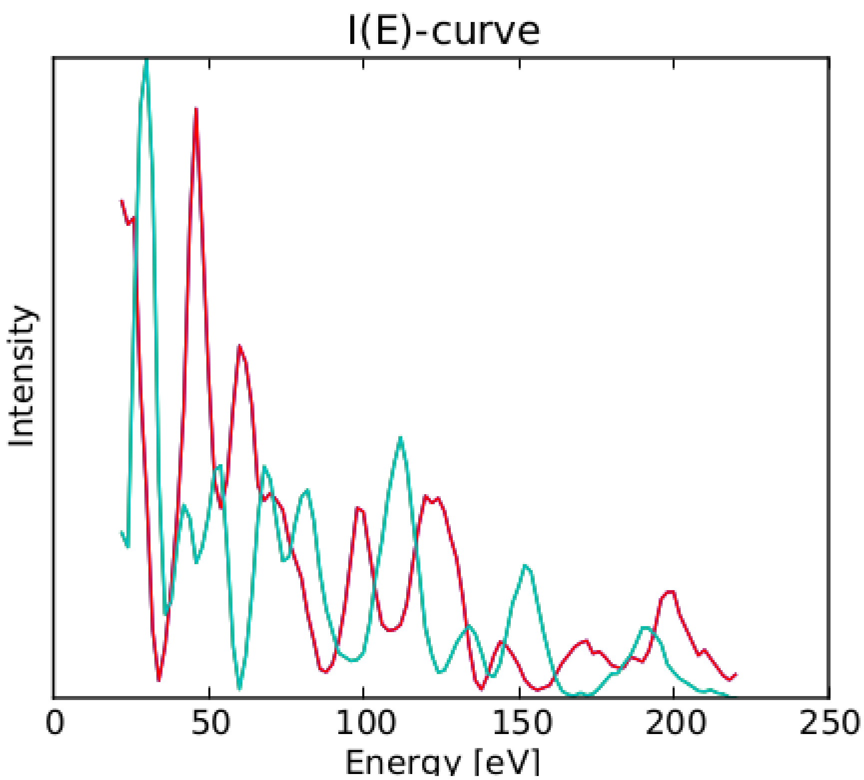

Fig.2. Intensity-energy spectra from easyLEED

The plotting function gets the intensity and energy data and uses the matplotlib plot function to do the actual plot. The intensity data is contained in a list of lists so every new spot to be followed adds its own list to the main list of the intensities. It is the same with the energy; the energy list contains as many lists as there is followed spots even though all of the spots have the same energies. So the plotting function uses a simple for loop to go through and plot the data. Some labels and a title are also added to the plot by the function. An example plot can be seen in Figure 2 and Figure 3 shows the PyQt4 widget drawing the plot.

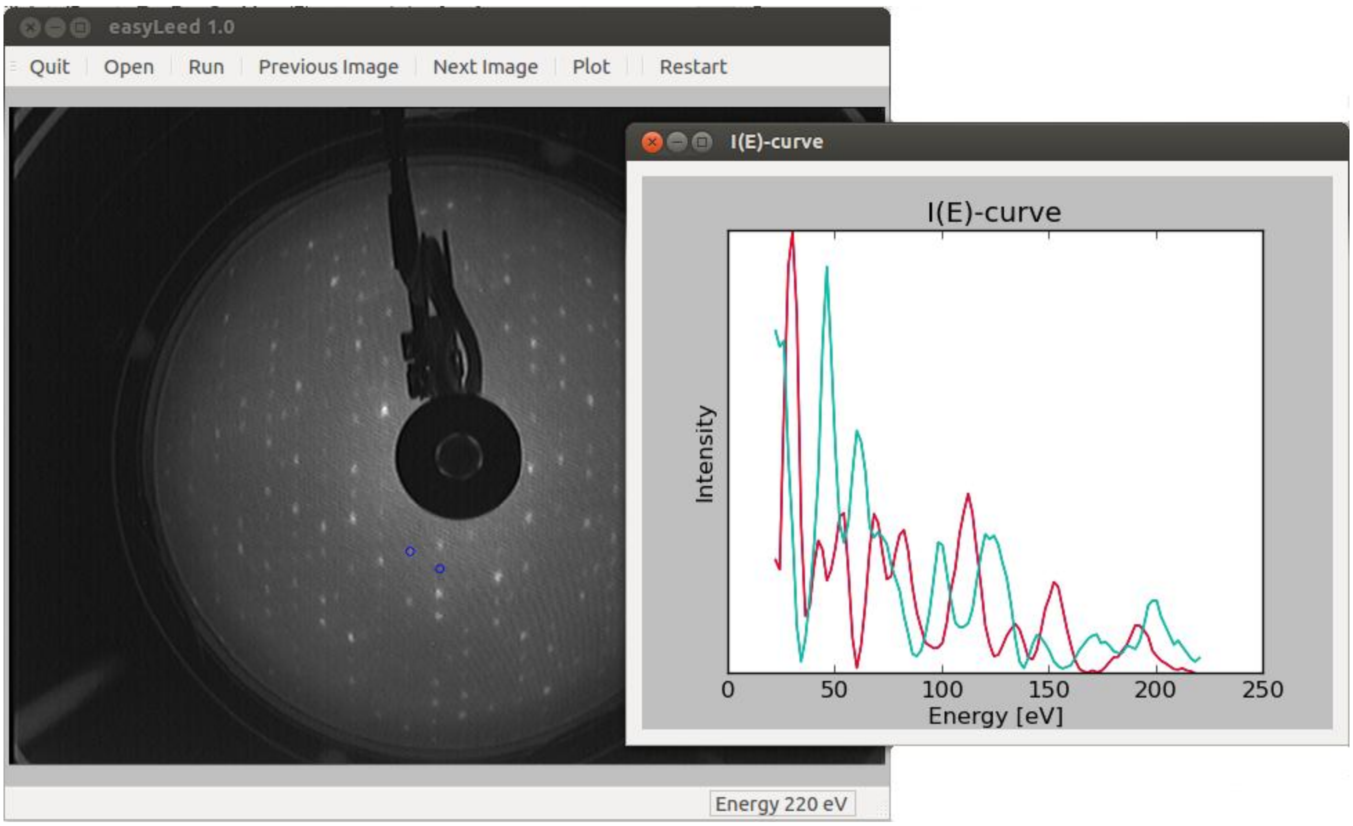

Fig.3. PyQt4 widget with embedded matplotlib

Plotting the average of the intensities is in a function of its own. It takes the same intensity-energy data as the normal plotting function, flattens the list containing all the intensities and calculates the average of these intensities. The function adds the same labels and title to the plot as the normal plotting function. The average plot is thicker than the normal ones as Figure 4 shows. This helps to identify it if it is drawn at the same time as the other intensities.

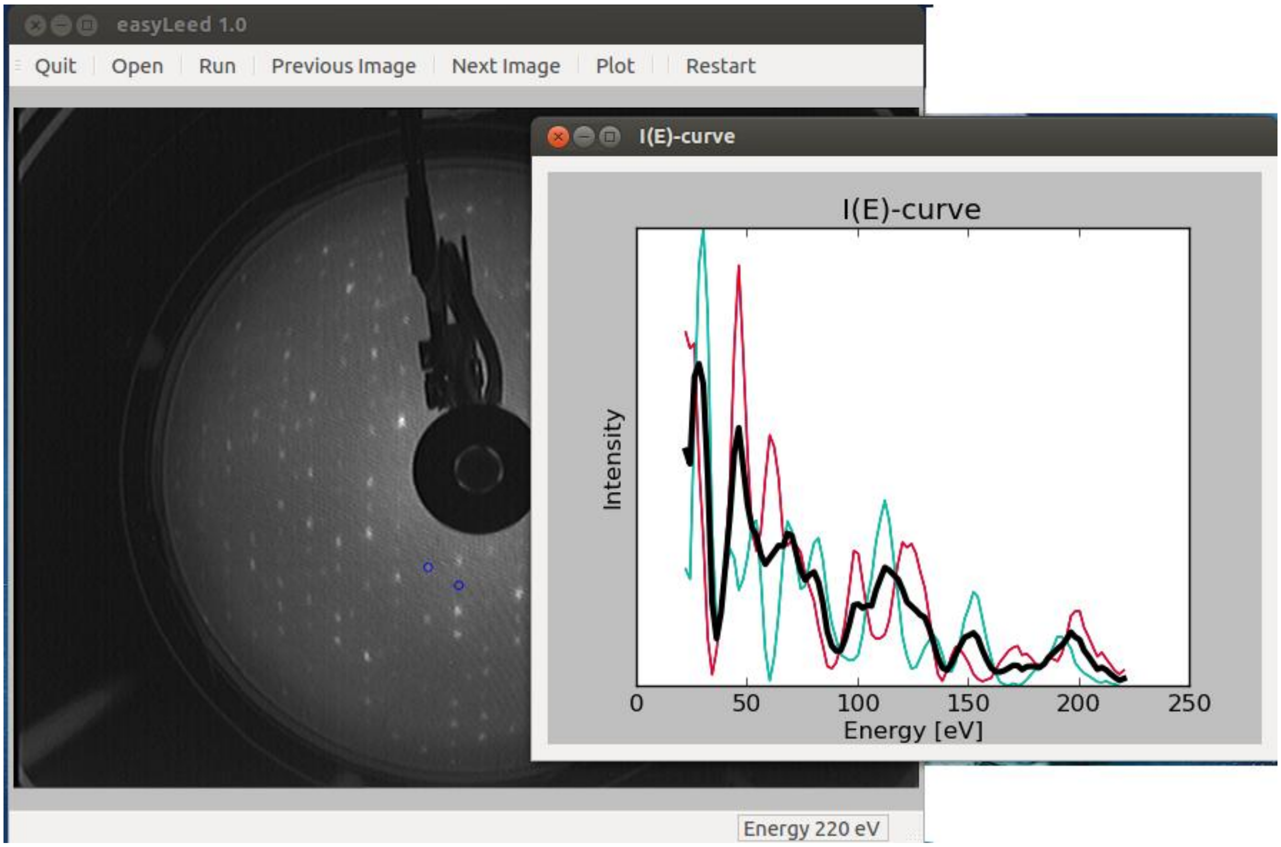

Fig.4. A plot with the basic intensities and the average intensity

The plotting method, i.e. to plot the intensities, plot the average and plot the intensities and the average at the same time, is chosen from a widget shown in Figure 5. The function executed when pressing the accept button runs the relevant functions based on the checked radio button. The two plotting functions are separate functions mainly because of this even though they both contain some identical code, for example, the labeling. So either function can be run on its own or both of them can be run simultaneously as the needs of the researchers vary, some may only need the average intensity while others need the whole package.

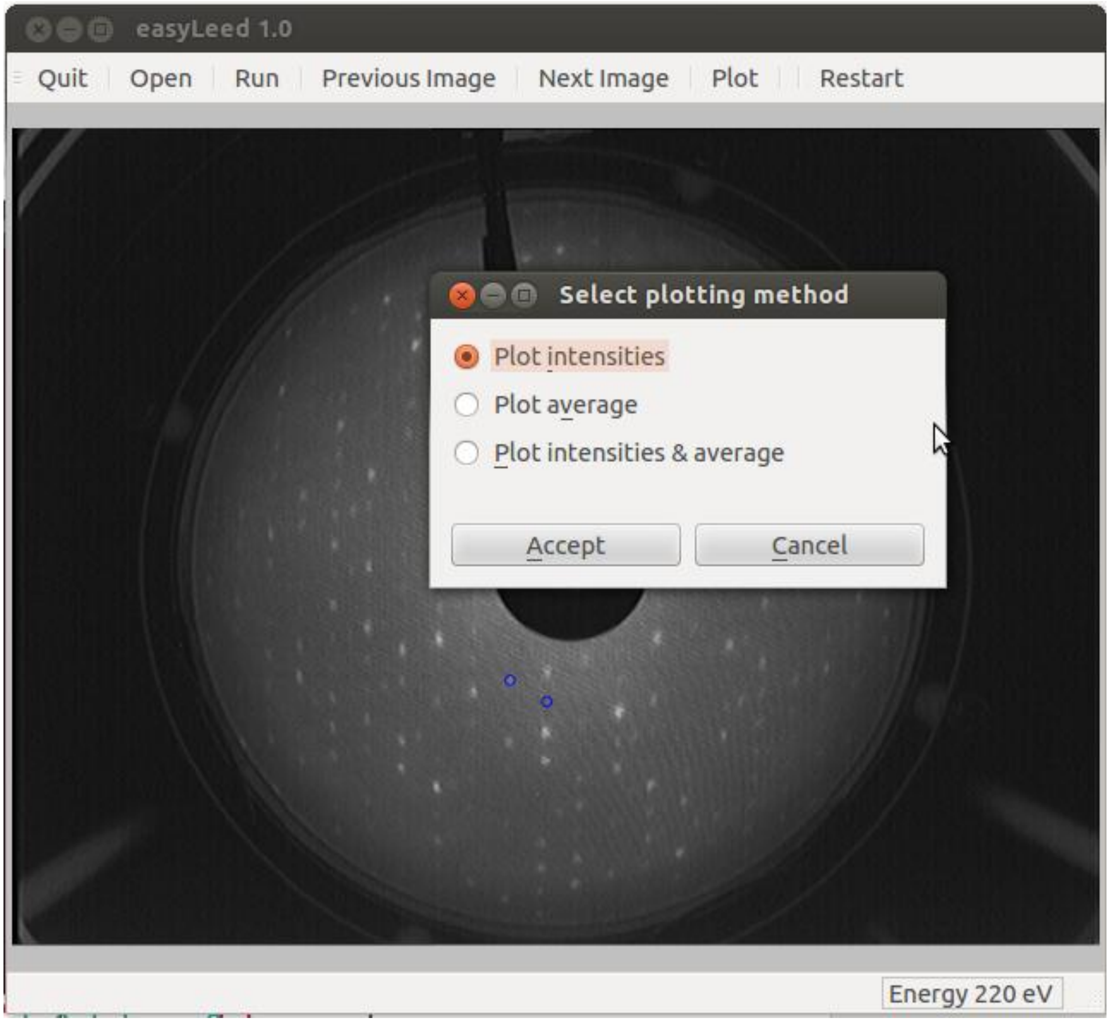

Fig.5. Selecting a plotting method

4.2 Parameter widget

Tracking parameters such as the Kalman tracker process noise have an impact on the correctness of the tracked intensities so it is important for the user to have the ability to change the values to their liking. The default values are saved in a config module so the user would need to change the actual file and this might be bothersome and time-consuming for the basic user. A widget was added so the user can easily change these parameters.

The parameter widget shown in Figure 6 fetches the values from the config module when the module is loaded to the program. At the current style it is not possible to change the hard values so every time the program starts the values are set to the default ones. To circumvent this the possibility to save and load parameter values was added. This is not a very desirable feature and there would probably be a better way of doing this but to prevent some major changes in the structure of the program this was far the best option at hand. To think positive an inexperienced user can now easily reload to default values if the user set values are not working.

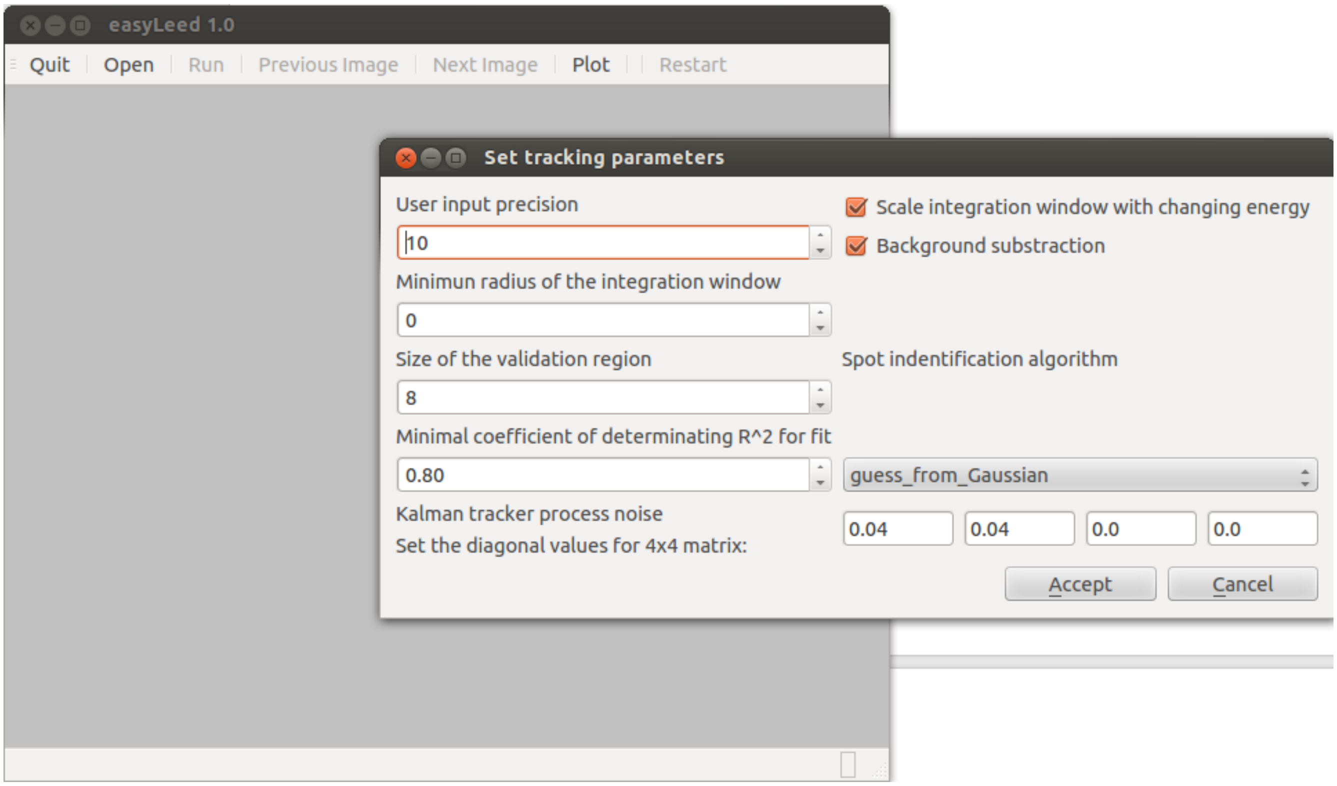

Fig.6. The tracking parameter widget

4.3 Export/Import

EasyLEED has an option to export and import several things. The most useful of them from the researchers point of view is the possibility to export the intensity-energy spectra. It is done with the matplotlib's built in export function.

The initial spot locations can be saved and loaded back. The save locations function actually saves the spot locations on all the energies but due to prioritising and to lack of time the load function only loads and adds the initial spot locations to the program. So the user has to run the tracking algorithm again if he/she wants to have the energy/intensity spectra. The location saving and loading was added a manual follow option in mind. Usually the tracking is done by hand and the person doing the tracking selects the followed spot on every energy so some users might find this manual option preferable to the automatic tracking. This is definitely a part that still needs some work in the future to be any use in research but it's not really a part of this thesis.

In addition to location saving and loading the parameters used in the tracking can be saved and loaded. Due to the default values being in a different module the values are reset to default every time the config module is loaded to the program. So the tracking values can only be set to the session currently running so the user might want to save and load some commonly used parameter values to maximize the correctness of the tracked intensities.

4.4 Quick access, documentation

For the user to quickly access the most used features a toolbar has been added to the program. The toolbar can be seen from Figures 3, 4 and 5. It is at the top of the program screen under the menu bar and contains the most used functionalities: opening image files, running the tracking algorithm, plotting the intensity and resetting the program and the images to the original state. The toolbar also contains a quit button and the buttons to browse the pictures so the user can easily choose the wanted starting energy for the tracking. The toolbar aims to have the most used functionalities under one click so the program is quick to use and no time is spent browsing through the menu for the right functionality.

The documentation and user guides are done in sphinx opposed to the original plan of plain text documents. Sphinx is a tool designed to easily generate documentation of Python code with a support also for C and C++. It converts reStructuredText files into HTML (Hypertext Markup Language) websites and other formats including PDF, EPub (electronic publication) and man, an unix manual format. [2] Sphinx was chosen because it can easily be used to make both PDF and HTML documentation and it can fetch the class documentation from the actual source code. An example of the Sphinx generated HTML documentation can be seen in Figure 7. The user guides include a short step-by-step guide how to quickly get the program calculating the intensities and a longer, more detailed guide on how the program works.

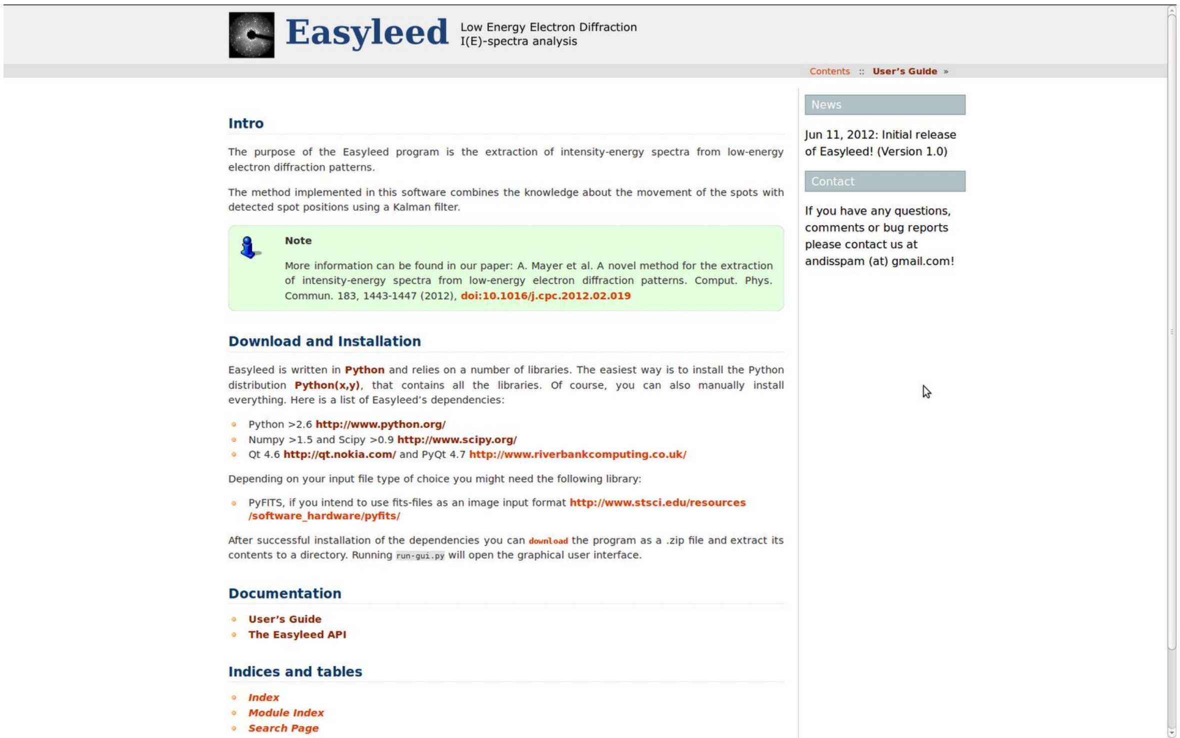

Fig.7. Sphinx generated HTML documentation of easyLEED, old revision

5. Analysis

All the things that were planned in the beginning were added to the program. Additional features like setting the tracking parameters were also introduced. All these functionalities increase the usefulness of easyLEED in their part. Of course there is always some room for improvement and for example the tracking still needs an improved algorithm as it tends to sometimes lose the spots it's supposed to be following unfortunately it was also a part that was ruled out of this thesis. Correctly working tracking is the thing that would make easyLEED really useful in research.

There was some excess work on the plotting functions that could have been prevented with a better research on the subject. Redoing the plotting took some extra time that could have been used in other parts of the work. For example the manual follow of the spots would have been a great addition to the program's functionality but it is unfortunately something that someone else needs to add to the program in the future.

The main part is now to make easyLEED available for the research community. It is still underway but should be done by the end of this year but it is done as a part of my work at the university rather than a part of this thesis. Using Sphinx in the documentation has an advantage of an easily made HTML file and the research group this thesis was done in has a website where the program can be eventually published.

References

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

[2] Brandl, Georg, *Sphinx Python Documentation Generator*, URL: http://sphinx.pocoo.org/index.html (13.8.2012)

[3] Gavaza, G. M, Yu, Z.X., Tsang, L., Chan, C.H., Tong, S.Y., Van Hove, M.A., *Theory of low-energy electron diffraction for detailed structural determination of nanomaterials: Ordered structures*, Phys. Rev. B 75 (2007) 014114.

[4] Gavaza, G. M, Yu, Z.X., Tsang, L., Chan, C.H., Tong, S.Y., Van Hove, M.A., *Theory of low-energy electron diffraction for detailed structural determination of nanomaterials: Finite-size and disordered structures,* Phys. Rev. B 75 (2007) 235403.

[5] Giergiel, J., Hopster, H., Lawrence, J.M., Hemminger, J.C., Kirschner, J., Rev. Sci., *A video based spin polarized LEED data acquisition system*, Instrum. 66 (6) (1995) 3475–3479.

[6] Heinz, K., Rep., *LEED and DLEED as modern tools for quantitative surface structure determination,* Prog. Phys. 58 (1995) 637.

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

[8] Hunter, John, Dale, Darren, Droettboom, Michael, *matplotlib: python plotting — Matplotlib v1.1.0 documentation*, URL: http://matplotlib.sourceforge.net/ (31.5.2012)

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

[10] 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.

[11] 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*, Computer Physics Communications, 183 (2012), 7, pages 1443–1447

[12] Moritz, W., J. Phys. C, *Effective calculation of LEED intensities using symmetry-adapted functions*, Solid State Phys. 17 (1984) 353.

[13] Moritz, W., Landskron, J., Deschauer, *Perspectives for surface structure analysis with low energy electron diffraction*, M., Surf. Sci. 603 (2009) 1306–1314.

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

[15] Ogletree, D.F., Blackman, G.S., Hwang, R.Q., Starke, U., Somorjai, G.A., Katz, J.E., *A new pulse counting low- energy electron diffraction system based on a position sensitive detector*, Rev.Sci. Instrum. 63 (1992) 104–113.

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

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

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

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

[20] Soares, E.A., de Castilho, C.M.C., de Carvalho, V.E., *Advances on surface structural determination by LEED*, J. Phys. Condens. Matter 23 (2011) 303001.

[21] Van Hove, M.A., Moritz, W., Over, H., Rous, P.J., Wander, A., Barbieri, A., Materer, N., Starke, U., Somorjai, G.A., *Automated determination of complex surface structures by LEED*, Surf. Sci. Rep. 19 (3–6) (1993) 191–229.

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