Module Guide (MG)
STEMMoireRec

Alexandre Pofelski
github: slimpotatoes

December 14, 2021

1 Revision History

Table 1: **Revision History**

Date	**Version**	**Notes**
13/12/2021	1.0	First Draft

Contents

List of Tables

List of Figures

2 Module Hierarchy

This section provides an overview of the module design. Modules are summarized in a hierarchy decomposed by secrets in table 2. The modules listed below, which are leaves in the hierarchy tree, are the modules that will actually be implemented.

M 1 User input Module (described in section 4.1.1)

M 2 Crystal Module (described in section 4.1.2)

M 3 Moiré simulation Module (described in section 4.1.3)

M 4 Moiré recovery Module (described in section 4.1.4)

M 5 Output Module (described in section 4.1.5)

M 6 Data Structure Module (described in section 4.2.1)

M 7 Mask Module (described in section 4.1.6)

M 8 Fourier Transform Module (described in section 4.2.2)

M 9 Moiré correction Module (described in section 4.1.7)

M 10 Plotter Module (described in section 4.2.3)

Level 1	**Level 2**
Behaviour-Hiding Module	User input Crystal Moiré simulation Moiré recovery Mask Moiré correction Output
Software Decision Module	Fourier Transform Plotter Data structure

Table 2: Module Hierarchy

3 Connection Between Requirements and Design

The design of the system is intended to satisfy the requirements developed in the SRS. In this stage, the system is decomposed into modules. The connection between requirements and modules is listed in Table 3.

This section shows two traceability matrices: between the modules and the requirements and between the modules and the anticipated changes.

Req.	**Modules**
R 1	M 1
R 2	M 1, M 2
R 3	M 1
R 4	M 1
R 5	M 1
R 6	M 3, M 5
R 7	M 1
R 8	M 4, M 5
R 9	M 1, M 4
R 10	M 4, M 5

Table 3: Trace Between Requirements and Modules

4 Module Decomposition

Modules are decomposed according to the principle of "information hiding" proposed by [**?**]. The *Secrets* field in a module decomposition is a brief statement of the design decision hidden by the module. The *Services* field specifies *what* the module will do without documenting *how* to do it. For each module, a suggestion for the implementing software is given under the *Implemented By* title. If the entry is *OS*, this means that the module is provided by the operating system or by standard programming language libraries. Also indicate if the module will be implemented specifically for the software.

Only the leaf modules in the hierarchy have to be implemented. If a dash (–) is shown, this means that the module is not a leaf and will not have to be implemented. Whether or not this module is implemented depends on the programming language selected.

4.1 Behaviour-Hiding Module

4.1.1 Input Module (M 1)

Secrets: The format and structure of the input data.

Services: Loads, verifies and stores the input data into the appropriate data and object structure.

Implemented By: STEMMoireRec

4.1.2 Mask Module (M 2)

Secrets: All the crystalline information with respect chosen by the user.

Services: Generate all the required information regarding the crystal structure : the crystal base, the allowed reflections and the projection in sampling base

Implemented By: STEMMoireRec

4.1.3 SMH simulation Module (M 3)

Secrets: Algorithm to simulate the Moiré process on the crystalline reflections.

Services: Simulates the Moiré process in Fourier space using the pixel size p of the I_{SMH}

Implemented By: STEMMoireRec

4.1.4 Moiré recovery Module (M 4)

Secrets: Algorithm to reconstruct the oversampled electron micrograph from the STEM Moiré hologram.

Services: Performs the necessary elements for the reconstruction : seperation, correction, addition and inverse Fourier transform.

Implemented By: STEMMoireRec

4.1.5 Output(M 5)

Secrets: Figure displaying the necessary information for the user..

Services: Display the intermediate and final results to the user.

Implemented By: STEMMoireRec

4.1.6 Mask Module (M 7)

Secrets: Algorithm to isolate a portion of an image.

Services: Isolates and weights a sub area of an image by fixing the parameter M_j.

Implemented By: STEMMoireRec

4.1.7 Moiré correction Module (M 9)

Secrets: Algorithm to convert a Moiré wave vector to a crystalline wave vector.

Services: Performs the affine vectorial transformation using the Moiré wave vectors and the sampling vectors as inputs.

Implemented By: STEMMoireRec

4.2 Software Decision Module

4.2.1 Data Structure Module (M 6)

Secrets: Data format for the STEMMoireRec object.

Services: Provides convenient format to store, read and manipulate all elements related to the STEMMoireRec data structure..

Implemented By: Python library

4.2.2 Fourier Transform Module (M 8)

Secrets: Fourier transform algorithm.

Services: Transforms data into its Fourier transform.

Implemented By: Python library

4.2.3 Generic GUI/Plot Module (M 10)

Secrets: Display data to user using plotting solutions

Services: Provides the methods to display the data to the user

Implemented By: Python library

5 Use Hierarchy Between Modules

In this section, the uses hierarchy between modules is provided. [**?**] said of two programs A and B that A *uses* B if correct execution of B may be necessary for A to complete the task described in its specification. That is, A *uses* B if there exist situations in which the correct functioning of A depends upon the availability of a correct implementation of B. Figure 1 illustrates the use relation between the modules. It can be seen that the graph is a directed acyclic graph (DAG). Each level of the hierarchy offers a testable and usable subset of the system, and modules in the higher level of the hierarchy are essentially simpler because they use modules from the lower levels.

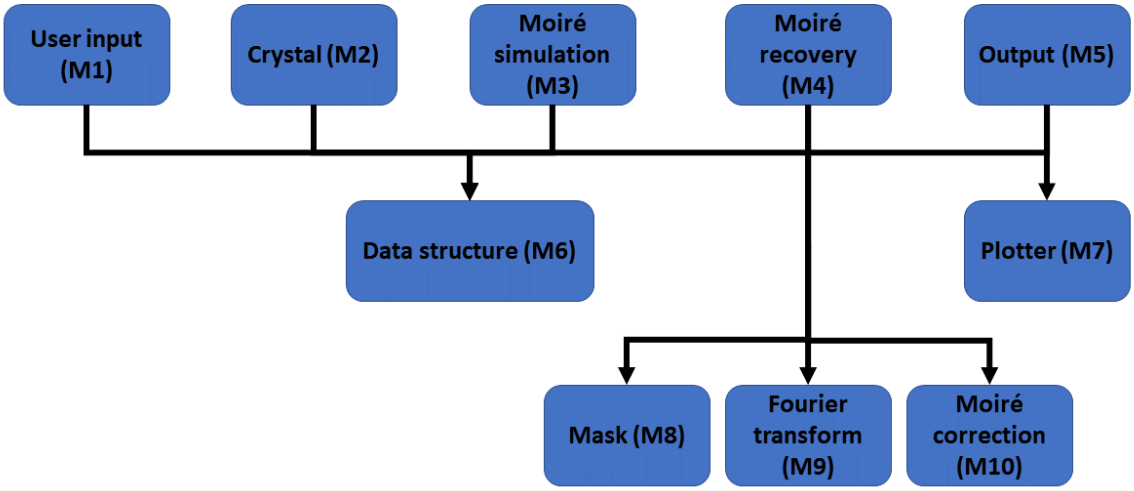

Figure 1: Use hierarchy among modules