Test Plan:
STEM Moiré GPA

Alexandre Pofelski
macid: pofelska
github: slimpotatoes

November 7, 2017

1 Revision History

Table 1: **Revision History**

Date	**Version**	**Notes**
23/10/2017	1.0	First Draft

2 Symbols, Abbreviations and Acronyms

Symbols, Abbreviations and Acronyms used in the Test Plan document are regrouped under the following table, in section 2.2 and in section 2.3 of the SRS document.

symbol	**description**
GUI	Graphical Unit Interface
N/A	Non-Applicable

Contents

1 Revision History **i**

2 Symbols, Abbreviations and Acronyms **ii**

3 General Information **1**
- 3.1 Purpose . . . 1
- 3.2 Scope . . . 1

4 Plan **2**
- 4.1 Software Description . . . 2
- 4.2 Test Team . . . 2
- 4.3 Automated Testing Approach . . . 2
- 4.4 Verification Tools . . . 2
- 4.5 Non-Testing Based Verification . . . 2

5 System Test Description **3**
- 5.1 Tests for Functional Requirements . . . 3
  - 5.1.1 Output Verfication (and/or Validation) tests . . . 3
- 5.2 Tests for Nonfunctional Requirements . . . 7
  - 5.2.1 Evolution of precision with noise in STEM Moiré GPA 7
  - 5.2.2 Evolution of the time calculations with data size . . . 8
  - 5.2.3 Evolution of the accuracy of STEM Moiré GPA with environment (reproducibility) . . . 9

6 Unit Testing Plan **11**
- 6.1 Input Verification test . . . 11
- 6.2 Output Verification test . . . 15

7 Traceability: Tests Cases vs Requirements **18**

8 Prioritization **19**

List of Tables

1 **Revision History** . . . i

2 **Traceability table: Test vs Requirements** [Capitalize Table —SS] 19
3 **Priority table** [Capitalize Table —SS] 20

3 General Information

3.1 Purpose

The purpose of the document is to provide the plan for testing STEM Moiré GPA software with respect to the requirements specified in the SRS document. [An explicit web-link to your GitHub repo would be nice. —SS]

3.2 Scope

The scope of the test plan is described below:

- STEM Moiré GPA will be written in Python.
- The proposed test plan is focusing on system and unit testing to verify the functional and non functional requirements of STEM Moiré GPA. Because of the nature of IM 1, IM 2 and IM 3 (see SRS document), a complete verification of their respective outputs is not possible. In these cases, the system tests are designed as validation tests (including algorithm errors) and will not catch a code error [Do you mean "will not distinguish a code error for a modelling error"? —SS]. A physical understanding of the test output is recommended to interpret properly its quantitative aspect, since an assessment by the test team on the quality of the test results is required to validate it (reasonable error?). If the error observed is unreasonably high, an error in the code tested might be postulated.
- Regarding the numerous requirements of STEM Moiré GPA and their relative complexity to be fully verified, only a portion of STEM Moiré GPA is proposed to be tested. A specific focus is proposed on key aspects that are identified as the highest potential source of errors. The testing of the GUI won't be approached in this document but should be definitely considered in the future.

[The text is better for version control, and for reading in other editors, if you use a hard-wrap at 80 characters —SS]

4 Plan

Based on the scope of the test plan, only R 2, R 3, R 6, R 7, R 9, R 10, R 12, R 13, NR 1, NR 2, NR 8 and NR 9 are planned to be verified and/or validated.

[Nice to see the external references! —SS]

4.1 Software Description

STEM Moiré GPA software is converting a STEM Moiré hologram into deformation maps. Details on the goal and the requirements of STEM Moiré GPA are provided in the Problem Statement and the SRS documents.

4.2 Test Team

The author of the document is the only member of the test team.

4.3 Automated Testing Approach

By predefining some user inputs, an automated testing approach can be considered. The unit tests and most of the system tests will be executed and validated automatically. Some system tests (Test 2, Test 3, Test 4, Test 6, Test 7, Test 8) require at least a manual assessment of the test results and are thus removed from the automated scope.

4.4 Verification Tools

With respect to the limited experience of the test team regarding verification and validation, only a unit test framework is going to be used as verification tools. **Pytest**, a Python unit test framework, will be used to test STEM Moiré GPA software.

4.5 Non-Testing Based Verification

N/A[Possible to put Isobel to go through the code but I don't want to add it in this document since the code inspection is a favor and not part of the plan :). —Author] [Okay. That sound fair. —SS]

5 System Test Description

5.1 Tests for Functional Requirements

5.1.1 Output Verfication (and/or Validation) tests

Test R 3 in IM 1: Correctness of the sampling theory application when undersampling g

Test 1 Test-aliasing-frequency-simple

- ⋆ Type: Functional, Dynamical, Automated
- ⋆ Initial State: N/A
- ⋆ Input: $I_{C_{\mathrm{ref}}} = e^{2i\pi gx}$, p such that $\overrightarrow{q} = \frac{1}{p}\vec{u_x}$
- ⋆ Expected output : $\widetilde{I}_{SHM_{\mathrm{sim}}} = \delta(\vec{\nu} - \vec{q})$
- ⋆ Output: $\widetilde{I}^{t}_{SHM_{\mathrm{sim}}}$ to be tested as a Dirac delta function at $\overrightarrow{q}$

The position in frequency space of the aliased frequency (transformation of one $\overrightarrow{g_j}^{C_{\mathrm{ref}}}$ from $I_{C_{\mathrm{ref}}}$ to one $\overrightarrow{g_j}^{M_{\mathrm{ref}}}$ of $I_{SMH_{\mathrm{sim}}}$) is key to identify the vector $\overrightarrow{q_{n_j,m)j}}$. Test 1 is designed to check if the most basic transformation when under sampling is correctly performed. More complex versions of Test 1 can be designed with reference functions having, for example, a continuous description in Fourier space. Nevertheless, the more complex tests will cover way more cases than what $I_{C_{\mathrm{ref}}}$ typically is (a sparse function). Depending on the time availability, a more complete version of Test 1 will be designed.

Test R 7 in IM 2: Correctness of the GPA method application

Test 2 Test-Phase-Extraction-No-Strain

- ⋆ Type: Functional, Dynamical, Automated
- ⋆ Initial State: N/A
- ⋆ Input: $I_{SMH_{exp}} = e^{2i\pi gx}$, Mask M of one pixel at $g\overrightarrow{u_x}$ in $\widetilde{I}_{SMH_{exp}}$
- ⋆ Expected output $P_{\Delta\overrightarrow{g_j}^{M_{exp}}} = 0$, $\Delta\overrightarrow{g_j}^{M_{exp}} = \overrightarrow{0}$,

- $\star$ Test output:$P_{\Delta\overrightarrow{g_j}^{M_{exp}}}{}^t$, $\Delta\overrightarrow{g_j}^{M_{exp}t}$
  - $\forall\vec{r}\in\mathbb{I},\ E_{P_{\Delta\overrightarrow{g_j}^{M_{exp}}}}(\vec{r}) = |P_{\Delta\overrightarrow{g_j}^{M_{exp}}}{}^t(\vec{r})|$
  - $\forall\vec{r}\in\mathbb{I},\ E_{\Delta\overrightarrow{g_j}^{M_{exp}}}(\vec{r}) = |\Delta\overrightarrow{g_j}^{M_{exp}t}(\vec{r})|$

Test 3 Test-Phase-Extraction-Known-Strain

- $\star$ Type: Functional, Dynamical, Manual
- $\star$ Initial State: N/A
- $\star$ Input: $I_{SMH_{exp}} = e^{2i\pi(g+K(x))x}$, Mask M centred on $g\overrightarrow{u_x}$ in $\widetilde{I}_{SMH_{exp}}$ and with the minimum radius to include $K(x)$.
- $\star$ Expected output $P_{\Delta\overrightarrow{g_j}^{M_{exp}}} = K(x)x$, $\Delta\overrightarrow{g_j}^{M_{exp}} = K(x)\overrightarrow{u_x}$,
- $\star$ Test output:$P_{\Delta\overrightarrow{g_j}^{M_{exp}}}{}^t$, $\Delta\overrightarrow{g_j}^{M_{exp}t}$
  - $\forall\vec{r}\in\mathbb{I},\ E_{P_{\Delta\overrightarrow{g_j}^{M_{exp}}}}(\vec{r}) = |P_{\Delta\overrightarrow{g_j}^{M_{exp}}}{}^t(\vec{r}) - P_{\Delta\overrightarrow{g_j}^{M_{exp}}}(\vec{r})|$
  - $\forall\vec{r}\in\mathbb{I},\ E_{\Delta\overrightarrow{g_j}^{M_{exp}}}(\vec{r}) = |\Delta\overrightarrow{g_j}^{M_{exp}t}(\vec{r}) - \Delta\overrightarrow{g_j}^{M_{exp}}(\vec{r})|$

Test 4 Test-Phase-Extraction-Mask

- $\star$ Type: Functional, Dynamical, Manual
- $\star$ Initial State: N/A
- $\star$ Input: $I_{SMH_{exp}} = e^{2i\pi(g+H(x))x}$, Mask M centred on $g\overrightarrow{u_x}$ in $\widetilde{I}_{SMH_{exp}}$ with different radius ϵ.
- $\star$ Expected output $P_{\Delta\overrightarrow{g_j}^{M_{exp}}} = H(x)x$, $\Delta\overrightarrow{g_j}^{M_{exp}} = H(x)\overrightarrow{u_x}$,
- $\star$ Test output:$P_{\Delta\overrightarrow{g_j}^{M_{exp}}}{}^t$, $\Delta\overrightarrow{g_j}^{M_{exp}t}$
  - $\forall\vec{r}\in\mathbb{I},\ E_{P_{\Delta\overrightarrow{g_j}^{M_{exp}}}}(\vec{r},\epsilon) = |P_{\Delta\overrightarrow{g_j}^{M_{exp}}}{}^t(\vec{r},\epsilon) - P_{\Delta\overrightarrow{g_j}^{M_{exp}}}(\vec{r})|$
  - $\forall\vec{r}\in\mathbb{I},\ E_{\Delta\overrightarrow{g_j}^{M_{exp}}}(\vec{r},\epsilon) = |\Delta\overrightarrow{g_j}^{M_{exp}t}(\vec{r},\epsilon) - \Delta\overrightarrow{g_j}^{M_{exp}}(\vec{r})|$

Because the GPA method is itself based on an approximation (see IM 2 in SRS document), errors from the algorithm are added to the errors from the code. Both are probed at the same time in the tests and cannot be fully separated. Test 2 is probing the accuracy and the precision in the most basic case possible (no strain) and is a first indicator of the IM 2 performance. Test 3 and Test 4 are closer to a real case scenario and highlight a more realistic version of IM 2 performance. To get quantitative trends, the error will be characterized as a function of the mask properties (such as the radius) and the deformation magnitude. This characterization will allow the GPA algorithm of STEM Moiré GPA to be compared with other software using the same GPA algorithm. Another interest of the performance characterization is to see a quantitative effect of the user inputs on IM 2 and provide a "good practice guidance" in the user manual. [I like this idea! —SS]

Test R 10 in IM 3: Correctness of the unstrained reference calculation

Test 5 Test-constant-delta-g

- ⋆ Type: Functional, Dynamical, Automated
- ⋆ Initial State:
- ⋆ Input:
  - $\forall \vec{r} \in \mathbb{I},\ \Delta \overrightarrow{g}^{M_{\exp}}(\vec{r}) = \overrightarrow{C}$
  - U array of 1 pixel wherever on $\Delta \overrightarrow{g}^{M_{\exp}}(\vec{r})$
  - $\overrightarrow{g}^{M_{\exp}} = {g_x}^{M_{\exp}}\overrightarrow{u_x} + {g_y}^{M_{\exp}}\overrightarrow{u_y}$
- ⋆ Expected output: $\overrightarrow{g}^{M_{\exp}}_{\text{uns}} = \overrightarrow{g}^{M_{\exp}} - \overrightarrow{C}$, $\Delta\overrightarrow{g_j}^{M_{\exp}}_{\text{cor}} = \overrightarrow{0}$
- ⋆ Output: ${\overrightarrow{g}^{M_{\exp}}_{\text{uns}}}^t$, ${\Delta\overrightarrow{g_j}^{M_{\exp}}_{\text{cor}}}^t$
  - $E_{\overrightarrow{g}^{M_{\exp}}_{\text{uns}}} = || {\overrightarrow{g}^{M_{\exp}}_{\text{uns}}}^t - \overrightarrow{g}^{M_{\exp}}_{\text{uns}} ||$
  - $\forall \vec{r} \in \mathbb{I},\ E_{\Delta\overrightarrow{g_j}^{M_{\exp}}_{\text{cor}}} = ||{\Delta\overrightarrow{g_j}^{M_{\exp}}_{\text{cor}}}^t||$

Test 5 is not realistic but test the code directly. Increasing the size of U, [write as U —SS]should have absolutely no effect on the output.

Test 6 Test-constant-delta-g-with-noise

* Type: Functional, Dynamical, Manual (can be automated)
* Initial State:
* Input:
  - $\overrightarrow{N}$, 2D random noise
  - $\forall \vec{r} \in \mathbb{I},\ \Delta \overrightarrow{g}^{M_{\text{exp}}}(\vec{r}) = \overrightarrow{C} + \overrightarrow{N}$
  - U array of $n \times m$ pixels wherever on $\Delta \overrightarrow{g}^{M_{\text{exp}}}(\vec{r})$
  - $\overrightarrow{g}^{M_{\text{exp}}} = g_x{}^{M_{\text{exp}}}\overrightarrow{u_x} + g_y{}^{M_{\text{exp}}}\overrightarrow{u_y}$
* Expected output: $\overrightarrow{g}^{M_{\text{exp}}}_{\text{uns}} = \overrightarrow{g}^{M_{\text{exp}}} - \overrightarrow{C}$, $\Delta \overrightarrow{g_j}^{M_{\text{exp}}}_{\text{cor}} = \overrightarrow{0}$
* Output: $\overrightarrow{g}^{M_{\text{exp}}{}^{t}}_{\text{uns}}, \Delta \overrightarrow{g_j}^{M_{\text{exp}}{}^{t}}_{\text{cor}}$
  - $E_{\overrightarrow{g}^{M_{\text{exp}}}_{\text{uns}}} = ||\overrightarrow{g}^{M_{\text{exp}}{}^{t}}_{\text{uns}} - \overrightarrow{g}^{M_{\text{exp}}}_{\text{uns}}||$
  - $\forall \vec{r} \in \mathbb{I},\ E_{\Delta \overrightarrow{g_j}^{M_{\text{exp}}}_{\text{cor}}} = ||\Delta \overrightarrow{g_j}^{M_{\text{exp}}{}^{t}}_{\text{cor}}||$

Test 6 is more representative of a real case than Test 5 and will be use to characterize the evolution of the error with respect to the level of noise and the size of U. If the error is unreasonably high, an error in the code can be postulated. The evaluation of $\overrightarrow{g}^{M_{\text{exp}}}_{\text{uns}}$ is critical on the quantitative estimation of strain and rotation. [I like this test, but (as for several other tests), there are still decisions that could be made that you have postponed. Specifically, what are the values of m and n? If the data is difficult to specify, maybe you could point to a sample file in your repo that you are going to use? —SS]

Test 7 Test-varying-delta-g

* Type: Functional, Dynamical, Manual
* Initial State:
* Input:
  - $\forall \vec{r} \in \mathbb{I},\ \overrightarrow{g}^{M_{\text{exp}}}(\vec{r}) = \overrightarrow{C}(\vec{r})$

– U array of $n \times m$ pixels wherever on $\Delta\overrightarrow{g}^{M_{\exp}}(\vec{r})$
– $\overrightarrow{g}^{M_{\exp}} = g_x{}^{M_{\exp}}\overrightarrow{u_x} + g_y{}^{M_{\exp}}\overrightarrow{u_y}$

⋆ Expected output: $\overrightarrow{g}^{M_{\exp}}_{\text{uns}} = \overrightarrow{g}^{M_{\exp}} - F(\overrightarrow{C}(\vec{r}))$, $\Delta\overrightarrow{g_j}^{M_{\exp}}_{\text{cor}} = \overrightarrow{C}(\vec{r}) - F(\overrightarrow{C}(\vec{r}))$ where $F(\overrightarrow{C}(\vec{r}))$ is the best possible linear fit of $\overrightarrow{C}(\vec{r})$ in U.

⋆ Output: $\overrightarrow{g}^{M_{\exp}}_{\text{uns}}{}^{t}, \Delta\overrightarrow{g_j}^{M_{\exp}}_{\text{cor}}{}^{t}$

– $E_{\overrightarrow{g}^{M_{\exp}}_{\text{uns}}} = ||\overrightarrow{g}^{M_{\exp}}_{\text{uns}}{}^{t} - \overrightarrow{g}^{M_{\exp}}_{\text{uns}}||$
– $\forall\vec{r} \in \mathbb{I},\ E_{\Delta\overrightarrow{g_j}^{M_{\exp}}_{\text{cor}}} = ||\Delta\overrightarrow{g_j}^{M_{\exp}}_{\text{cor}}{}^{t} - \Delta\overrightarrow{g_j}^{M_{\exp}}_{\text{cor}}||$

Test 7 might be impossible to properly interpret but the idea is to see the effect an improper choice of an unstrained reference. If a non uniform deformation field is present in the reference, STEM Moiré GPA is trying to find the best linear fit. A solution is calculated whatever the precision of the fit. However, if the quality of the fit is poor, it could be possible that STEM Moiré GPA informs/warns the user of a potential problem in the reference chosen.

5.2 Tests for Nonfunctional Requirements

5.2.1 Evolution of precision with noise in STEM Moiré GPA

Test NR 8 and NR 9

Test 8 Test-Black-Box-STEM-Moiré-GPA

⋆ Type: Functional, Dynamical, Black box, , Manual (can be automated)

⋆ Initial State: All user inputs are predefined $(p, M_1, M_2, U, \overrightarrow{q_{n_1,m_1}}, \overrightarrow{q_{n_2,m_2}})$ except the STEM Moiré hologram

⋆ Input: STEM Moiré holograms with different level of noise (from no noise to high level of noise):

– $I_{SMH_{\exp}} = I(\vec{r}) + N(\vec{r})$ with I a perfect SMH (with a defined strain field $\varepsilon(\vec{r})$ and rotation field $\omega(\vec{r})$) and N a random noise

⋆ Expected output $\forall\vec{r} \in \mathbb{I},\ \varepsilon(\vec{r})^t = \varepsilon(\vec{r}) \wedge \omega(\vec{r})^t = \omega(\vec{r})$

⋆ Output:

 – $\forall \vec{r} \in \mathbb{I},\ E_{\varepsilon}^{\text{STEM Moiré GPA}} = |\varepsilon(\vec{r})^t - \varepsilon(\vec{r})|$
 – $\forall \vec{r} \in \mathbb{I},\ E_{\omega}^{\text{STEM Moiré GPA}} = |\omega(\vec{r})^t - \omega(\vec{r})|$

Test 8 looks more like a validation test than a verification test. Nevertheless, if the error in Test 8 is unreasonably high, it is possible to consider the presence of a mistake in the code somewhere. On the validation side, Test 8 is highlighting the accuracy (without noise), and the precision (with noise) of STEM Moiré GPA seen by the end-user (code, user and algorithm errors included). The performance is then characterized as the error output function of the noise (or of the quality factor of $I_{SMH_{\text{exp}}}$ like the signal on noise ratio $SNR = \frac{I_{SMH_{\text{exp}}}}{N}$). The performance characterization can be used to see from which quality factor the accuracy and sensitivity of STEM Moiré GPA is too poor to consider the strain and rotation results reliable and warn the user in this case. The performance graph can also be as a base of comparison with another software (if it exists) doing the same data treatment.

[I like how you have used the words verification and validation consistently and correctly. —SS]

5.2.2 Evolution of the time calculations with data size

Test NR 2

Test 9 Test-stress-STEM-Moiré-GPA

⋆ Type: Functional, Dynimical, [spell check! —SS] Black box, Manual

⋆ Initial State: All user inputs are predefined $(p, M_1, M_2, U, \overrightarrow{q_{n_1,m_1}}, \overrightarrow{q_{n_2,m_2}})$ except the STEM Moiré hologram

⋆ Input: STEM Moiré holograms with different 2D array size.

⋆ Expected output $\forall \vec{r} \in \mathbb{I},\ \varepsilon(\vec{r})^t = \varepsilon(\vec{r}) \wedge \omega(\vec{r})^t = \omega(\vec{r})$ in a reasonable time frame [This could be less ambiguous. Specifying a specific time would not be much more meaningful, but maybe you could do a usability test? More realistically, you could just record the times for the tests and plot this data for others to consider. Rather than specify the performance, you are simply describing the performance as observed experimentally. —SS]

⋆ Output: Time to perform all the calculations

5.2.3 Evolution of the accuracy of STEM Moiré GPA with environment (reproducibility)

Test NR 1

Test 10 Test-Black-Box-STEM-Moiré-GPA-Linux

- ⋆ Type: Functional, Dynamical, Black box, Manual
- ⋆ Initial State: x64 Linux environment with all user inputs are predefined ($p, M_1, M_2, U, \overrightarrow{q_{n_1,m_1}}, \overrightarrow{q_{n_2,m_2}}$) except $I_{SMH_{\text{exp}}}$
- ⋆ Input: A perfect generated STEM Moiré hologram representing all the strain state in each quadrant $\mathbb{I}_i$ of the array $\mathbb{I}$ (unstrained in $\mathbb{I}_1$, 1% uniaxial strain along $\vec{u_x}$ and $\vec{u_y}$ in $\mathbb{I}_2$, 1% shear strain in $\mathbb{I}_3$, and 20 mrad rotation $\mathbb{I}_4$)
- ⋆ Expected output: In each quadrant
  - $\forall \vec{r} \in \mathbb{I}_1,\ \varepsilon(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \wedge \ \omega(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
  - $\forall \vec{r} \in \mathbb{I}_2,\ \varepsilon(\vec{r}) = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \wedge \ \omega(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
  - $\forall \vec{r} \in \mathbb{I}_3,\ \varepsilon(\vec{r}) = \begin{bmatrix} 0 & 0.01 \\ 0.01 & 0 \end{bmatrix} \wedge \ \omega(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
  - $\forall \vec{r} \in \mathbb{I}_4,\ \varepsilon(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \wedge \ \omega(\vec{r}) = \begin{bmatrix} 0 & 0.2 \\ -0.2 & 0 \end{bmatrix}$
- ⋆ Output: $\varepsilon(\vec{r})^t, \omega(\vec{r})^t$
  - $\forall \vec{r} \in \mathbb{I},\ E_{\varepsilon\ \text{Linux}}^{\text{STEM Moiré GPA}} = |\varepsilon(\vec{r})^t - \varepsilon(\vec{r})|$
  - $\forall \vec{r} \in \mathbb{I},\ E_{\omega\ \text{Linux}}^{\text{STEM Moiré GPA}} = |\omega(\vec{r})^t - \omega(\vec{r})|$

[Your outputs are unlikely to be an exact match. There will be some epsilon error. Maybe you could add a blanket statement at the start that points this out. Part of your output could be a summary of the observed errors (hopefully as relative errors, not absolute errors) for each test output in comparison to the ideal output. —SS]

Test 11 Test-Black-Box-STEM-Moiré-GPA-Mac

- ⋆ Type: Functional, Dynamical, Black box, Manual
- ⋆ Initial State: x64 Mac OS X environment with all user inputs are predefined $(p, M_1, M_2, U, \overrightarrow{q_{n_1,m_1}}, \overrightarrow{q_{n_2,m_2}})$ except $I_{SMH_{\text{exp}}}$
- ⋆ Input: A perfect generated STEM Moiré hologram representing all the strain state in each quadrant $\mathbb{I}_i$ of the array $\mathbb{I}$ (unstrained in $\mathbb{I}_1$, 1% uniaxial strain along $\vec{u_x}$ and $\vec{u_y}$ in $\mathbb{I}_2$, 1% shear strain in $\mathbb{I}_3$, and 20 mrad rotation $\mathbb{I}_4$)
- ⋆ Expected output: In each quadrant
  - $\forall \vec{r} \in \mathbb{I}_1,\ \varepsilon(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \wedge \vec{\omega(r)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
  - $\forall \vec{r} \in \mathbb{I}_2,\ \varepsilon(\vec{r}) = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \wedge \vec{\omega(r)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
  - $\forall \vec{r} \in \mathbb{I}_3,\ \varepsilon(\vec{r}) = \begin{bmatrix} 0 & 0.01 \\ 0.01 & 0 \end{bmatrix} \wedge \vec{\omega(r)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
  - $\forall \vec{r} \in \mathbb{I}_4,\ \varepsilon(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \wedge \vec{\omega(r)} = \begin{bmatrix} 0 & 0.2 \\ -0.2 & 0 \end{bmatrix}$
- ⋆ Output: $\varepsilon(\vec{r})^t, \omega(\vec{r})^t$
  - $\forall \vec{r} \in \mathbb{I},\ E^{\text{STEM Moiré GPA}}_{\varepsilon\ \text{Linux}} = |\varepsilon(\vec{r})^t - \varepsilon(\vec{r})|$
  - $\forall \vec{r} \in \mathbb{I},\ E^{\text{STEM Moiré GPA}}_{\omega\ \text{Linux}} = |\omega(\vec{r})^t - \omega(\vec{r})|$

Test 12 Test-Black-Box-STEM-Moiré-GPA-Windows

- ⋆ Type: Functional, Dynamical, Black box, Manual
- ⋆ Initial State: x64 Windows environment with all user inputs are predefined $(p, M_1, M_2, U, \overrightarrow{q_{n_1,m_1}}, \overrightarrow{q_{n_2,m_2}})$ except $I_{SMH_{\text{exp}}}$
- ⋆ Input: A perfect generated STEM Moiré hologram representing all the strain state in each quadrant $\mathbb{I}_i$ of the array $\mathbb{I}$ (unstrained in $\mathbb{I}_1$, 1% uniaxial strain along $\vec{u_x}$ and $\vec{u_y}$ in $\mathbb{I}_2$, 1% shear strain in $\mathbb{I}_3$, and 20 mrad rotation $\mathbb{I}_4$)

* Expected output: In each quadrant

  - $\forall \vec{r} \in \mathbb{I}_1,\ \varepsilon(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \wedge\ \omega(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
  - $\forall \vec{r} \in \mathbb{I}_2,\ \varepsilon(\vec{r}) = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \wedge\ \omega(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
  - $\forall \vec{r} \in \mathbb{I}_3,\ \varepsilon(\vec{r}) = \begin{bmatrix} 0 & 0.01 \\ 0.01 & 0 \end{bmatrix} \wedge\ \omega(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
  - $\forall \vec{r} \in \mathbb{I}_4,\ \varepsilon(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \wedge\ \omega(\vec{r}) = \begin{bmatrix} 0 & 0.2 \\ -0.2 & 0 \end{bmatrix}$

* Output: $\varepsilon(\vec{r})^t, \omega(\vec{r})^t$

  - $\forall \vec{r} \in \mathbb{I},\ E_{\varepsilon\ \text{Linux}}^{\text{STEM Moiré GPA}} = |\varepsilon(\vec{r})^t - \varepsilon(\vec{r})|$
  - $\forall \vec{r} \in \mathbb{I},\ E_{\omega\ \text{Linux}}^{\text{STEM Moiré GPA}} = |\omega(\vec{r})^t - \omega(\vec{r})|$

[Copying the same test 3 times in a row, with as far as I can tell the only difference being the OS is a maintenance headache. If you realize you have to change the test, you have to change it three times. A better version is to give the full test case once, and then the subsequence tests can say it is the same, with the one variability of OS changed. —SS]

[Technically x86 architecture should be also tested. —Author]

6 Unit Testing Plan

6.1 Input Verification test

Test R 2 in IM 1: Existence and Format of $I_{SMH_{\mathbf{exp}}}$, $I_{C_{\mathbf{ref}}}$ and p

Test 13 Test-Existence-SMH

* Type: Dynamical, Automated

* Initial State: Waiting for $I_{SMH_{\text{exp}}}$ user input

* Input: $I_{SMH_{\text{exp}}} = \emptyset$

- ⋆ Output: Error message $Err_{I_{SMH_{\mathrm{exp}}}}$ should match: "No STEM Moiré hologram, please load a proper image"

Test 14 Test-Format-SMH

- ⋆ Type: Dynamical, Automated
- ⋆ Initial State: Waiting for $I_{SMH_{\mathrm{exp}}}$ user input
- ⋆ Input: Various $I_{SMH_{\mathrm{exp}}}$ improper format
- ⋆ Output: Error message $Err_{I_{SMH_{\mathrm{exp}}}}$ should match: "Invalid STEM Moiré hologram format (expecting a 2D array)"

Test 15 Test-read-dm3-format

- ⋆ Type: Dynamical, Automated
- ⋆ Initial State: Waiting for $I_{SMH_{\mathrm{exp}}}$ or $I_{C_{\mathrm{ref}}}$ user input
- ⋆ Input: Pre defined random 2D array in dm3 format with specific tags (calibration, microscope, high tension, ...)
- ⋆ Output: Read the tag from the dm3 file and and test each tag collected with the pre defined tags

Test 16 Test-Existence-pixel

- ⋆ Type: Dynamical, Automated
- ⋆ Initial State: After importing $I_{SMH_{\mathrm{exp}}}$ and format validated
- ⋆ Input: p=$\emptyset$
- ⋆ Output: Error message Err_p should match: "No pixel size found"

Test 17 Test-Format-pixel

- ⋆ Type: Dynamical, Automated
- ⋆ Initial State: After importing $I_{SMH_{\mathrm{exp}}}$ and format validated
- ⋆ Input: Improper format of p

* Output: Error message Err_p should match: "Invalid pixel size"

Test 18 Test-Format-Reference

* Type: Dynamical, Automated
* Initial State: Waiting for $I_{C_{\text{ref}}}$ user input
* Input: Various $I_{C_{\text{ref}}}$ improper format
* Output: Error message $Err_{I_{C_{\text{ref}}}}$ should match: "Invalid Reference image format (expecting a 2D array)"

Test 19 Test-Existence-Reference

* Type: Dynamical, Automated
* Initial State: Waiting for $I_{C_{\text{ref}}}$ user input
* Input: $I_{C_{\text{ref}}} = \emptyset$
* Output: Error message $Err_{I_{C_{\text{ref}}}}$ should match: "No Reference image, please load a proper image"

Test R 6 in IM 2: Test of the mask user input

Test 20 Test-Existence-Mask

* Type: Dynamical, Automated
* Initial State: Waiting for M user input on $\widetilde{I}_{SMH_{\text{exp}}}$
* Input: M=$\emptyset$
* Output: Error message Err_M should match: "No Mask found"

Test 21 Test-Format-Mask

* Type: Dynamical, Automated
* Initial State: Waiting for M user input on $\widetilde{I}_{SMH_{\text{exp}}}$
* Input: M improper format (such as 1D array, 2D array out of bounds if

⋆ Output: Error message Err_M should match: "Improper mask format"

Test 22 Test-Position-Radius-Mask

⋆ Type: Dynamical, Automated

⋆ Initial State: Waiting for M user input on $\widetilde{I}_{SMH_{\text{exp}}}$

⋆ Input: Mask center $=(0,0)$, Mask radius=3

⋆ Output: Center of the circle mask positioned in the middle of the 2D array $\widetilde{I}_{SMH_{\text{exp}}}$ and with a radius of 3 pixels

Test R 9 in IM 3: Test of the unstrained region user input

Test 23 Test-Existence-U

⋆ Type: Dynamical, Automated

⋆ Initial State: Waiting for U user input on $P_{\Delta\overrightarrow{g_j}^{M_{exp}}}$

⋆ Input: U=$\emptyset$

⋆ Output: Error message Err_U should match: "No reference in phase image found"

Test 24 Test-Format-U

⋆ Type: Dynamical, Automated

⋆ Initial State: Waiting for U user input on $P_{\Delta\overrightarrow{g_j}^{M_{exp}}}$

⋆ Input: U improper format

⋆ Output: Error message Err_U should match: "Improper reference in phase image format"

Test 25 Test-Position-U

⋆ Type: Dynamical, Automated

⋆ Initial State: Waiting for U user input on $P_{\Delta\overrightarrow{g_j}^{M_{exp}}}$

⋆ Input:

- Position left top corner$=(10, 10)$,
- Position right bottom corner$=(30, 40)$.

⋆ Output: Rectangle center position $(20, 25)$, short length $l = 20$ pixels and long length $L = 30$ pixels

[GUI should be tested again here —Author]

6.2 Output Verification test

Test R 12 in IM 4: Affine vectorial transformation

Test 26 Test-Basic-affine-transformation

⋆ Type: Functional, Dynamical, Automated

⋆ Initial State: N/A

⋆ Input: $\overrightarrow{q_{n,m}} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$, $\overrightarrow{g_j}^{M_{\text{exp}}}_{\text{uns}} = \Delta\overrightarrow{g_j}^{M_{\text{exp}}}_{\text{cor}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $p = 3$

⋆ Output: $\overrightarrow{g_j}^{C_{\text{exp}}}_{\text{uns}} = \begin{bmatrix} 4 \\ -5 \end{bmatrix}$, $\Delta\overrightarrow{g_j}^{C_{\text{exp}}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Test 27 Test-Error-pixel-affine-transformation

⋆ Type: Functional, Dynamical, Automated

⋆ Initial State: Whatever $\overrightarrow{q_{n,m}}$, $\overrightarrow{g_j}^{M_{\text{exp}}}_{\text{uns}}$, $\Delta\overrightarrow{g_j}^{M_{\text{exp}}}_{\text{cor}}$

⋆ Input:$p = 0$ or $p = -10$

⋆ Output: Error message $Err_{p_{\text{affine}}}$ should match: "Invalid pixel size"

Test 28 Test-Error-q-affine-transformation

⋆ Type: Functional, Dynamical, Automated

⋆ Initial State: Whatever $\overrightarrow{q_{n,m}}$, $\overrightarrow{g_j}^{M_{\text{exp}}}_{\text{uns}}$, $\Delta\overrightarrow{g_j}^{M_{\text{exp}}}_{\text{cor}}$, and $p > 0$

⋆ Input:$\overrightarrow{q_{n,m}} = \begin{bmatrix} 1.3 \\ -1 \end{bmatrix}$ or $\overrightarrow{q_{n,m}} = \begin{bmatrix} 1 \\ -\sqrt{2} \end{bmatrix}$

⋆ Output: Error message $Err_{\vec{q}_{\text{affine}}}$ should match: "Invalid sampling vector $\overrightarrow{q_{n,m}}$"

TestR 13 in IM 5: Strain and rotation calculation

Test 29 Test-No-2D-strain

⋆ Type: Functional, Dynamical, Automated

⋆ Initial State: N/A

⋆ Input:

- $G_{\text{uns}}^{\text{exp}} = \begin{bmatrix} g_{1_{\text{uns}_x}}^{C_{\text{exp}}} & g_{2_{\text{uns}_x}}^{C_{\text{exp}}} \\ g_{2_{\text{uns}_x}}^{C_{\text{exp}}} & g_{2_{\text{uns}_y}}^{C_{\text{exp}}} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
- $\Delta G^{\text{exp}}(\vec{r}) = \begin{bmatrix} \Delta g_{1_x}^{C_{\text{exp}}}(\vec{r}) & \Delta g_{1_y}^{C_{\text{exp}}}(\vec{r}) \\ \Delta g_{2_x}^{C_{\text{exp}}}(\vec{r}) & \Delta g_{2_y}^{C_{\text{exp}}}(\vec{r}) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

⋆ Expected output:

- $\forall \vec{r} \in \mathbb{I},\ \varepsilon^{\text{exp}}(\vec{r}) = \begin{bmatrix} \varepsilon_{xx^{\text{exp}}}(\vec{r}) & \varepsilon_{xy^{\text{exp}}}(\vec{r}) \\ \varepsilon_{xy^{\text{exp}}}(\vec{r}) & \varepsilon_{yy^{\text{exp}}}(\vec{r}) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
- $\forall \vec{r} \in \mathbb{I},\ \omega^{\text{exp}}(\vec{r}) = \begin{bmatrix} 0 & \omega_{xy^{\text{exp}}}(\vec{r}) \\ -\omega_{xy^{\text{exp}}}(\vec{r}) & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

Test 30 Test-known-constant-2D-strain

⋆ Type: Functional, Dynamical, Automated

⋆ Initial State: N/A

⋆ Input:

- $G_{\text{uns}}^{\text{exp}} = \begin{bmatrix} g_{1_{\text{uns}_x}}^{C_{\text{exp}}} & g_{2_{\text{uns}_x}}^{C_{\text{exp}}} \\ g_{2_{\text{uns}_x}}^{C_{\text{exp}}} & g_{2_{\text{uns}_y}}^{C_{\text{exp}}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- Various $\Delta G^{\text{exp}}(\vec{r}) = \begin{bmatrix} \Delta g_{1_x}^{C_{\text{exp}}}(\vec{r}) & \Delta g_{1_y}^{C_{\text{exp}}}(\vec{r}) \\ \Delta g_{2_x}^{C_{\text{exp}}}(\vec{r}) & \Delta g_{2_y}^{C_{\text{exp}}}(\vec{r}) \end{bmatrix}$
  - Uniaxial strain along $\vec{u_x}$, $\Delta G^{\text{exp}}(\vec{r}) = \begin{bmatrix} 0.1 & 0 \\ 0 & 0 \end{bmatrix}$

* Uniaxial strain along $\vec{u_y}$, $\Delta G^{\text{exp}}(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0.1 \end{bmatrix}$
* Pure shear strain, $\Delta G^{\text{exp}}(\vec{r}) = \begin{bmatrix} 0 & 0.1 \\ 0.1 & 0 \end{bmatrix}$
* Pure rotation, $\Delta G^{\text{exp}}(\vec{r}) = \begin{bmatrix} 0 & 0.1 \\ -0.1 & 0 \end{bmatrix}$

⋆ Expected output:

- Uniaxial strain along $\vec{u_x}$,
$$\begin{cases} \forall \vec{r} \in \mathbb{I},\ \varepsilon^{\text{exp}}(\vec{r}) = \begin{bmatrix} -\frac{1}{11} & 0 \\ 0 & 0 \end{bmatrix} \\ \forall \vec{r} \in \mathbb{I},\ \omega^{\text{exp}}(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \end{cases}$$
- Uniaxial strain along $\vec{u_y}$,
$$\begin{cases} \forall \vec{r} \in \mathbb{I},\ \varepsilon^{\text{exp}}(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{11} \end{bmatrix} \\ \forall \vec{r} \in \mathbb{I},\ \omega^{\text{exp}}(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \end{cases}$$
- Pure shear strain,
$$\begin{cases} \forall \vec{r} \in \mathbb{I},\ \varepsilon^{\text{exp}}(\vec{r}) = \begin{bmatrix} 0 & -\frac{1}{11} \\ -\frac{1}{11} & 0 \end{bmatrix} \\ \forall \vec{r} \in \mathbb{I},\ \omega^{\text{exp}}(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \end{cases}$$
- Pure rotation,
$$\begin{cases} \forall \vec{r} \in \mathbb{I},\ \varepsilon^{\text{exp}}(\vec{r}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \forall \vec{r} \in \mathbb{I},\ \omega^{\text{exp}}(\vec{r}) = \begin{bmatrix} 0 & \frac{1}{11} \\ -\frac{1}{11} & 0 \end{bmatrix} \end{cases}$$

⋆ Output: $\varepsilon^{\text{exp}}(\vec{r})^t - \varepsilon^{\text{exp}}(\vec{r}), \omega^{\text{exp}}(\vec{r})^t - \omega^{\text{exp}}(\vec{r})$ and test them to be 0:

Test 31 <u>Test-Improper-ΔG^{exp}</u>

⋆ Type: Functional, Dynamical, Automated

⋆ Initial State: Whatever $\Delta G^{\text{exp}}(\vec{r})$

⋆ Input:

– $G^{\text{exp}}_{\text{uns}} = \begin{bmatrix} g^{C_{\text{exp}}}_{1_{\text{uns}_x}} & g^{C_{\text{exp}}}_{2_{\text{uns}_x}} \\ g^{C_{\text{exp}}}_{2_{\text{uns}_x}} & g^{C_{\text{exp}}}_{2_{\text{uns}_y}} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

⋆ Output: Error message $Err_{G^{\text{exp}}_{\text{uns}}}$ should match: "Matrix $G^{\text{exp}}_{\text{uns}}$ non invertible, the crystalline wave vectors are linked (collinear). Please chose [choose —SS] another combination of crystalline wave vectors."

Test 32 Test-Improper-$G^{\text{exp}}_{\text{uns}}$

⋆ Type: Functional, Dynamical, Automated

⋆ Initial State: N/A

⋆ Input:

– $\Delta G^{\text{exp}} + G^{\text{exp}}_{\text{uns}} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

⋆ Output: Error message $Err_{\Delta G^{\text{exp}}}$ should match: "Matrix G^{exp} non invertible. If $G^{\text{exp}}_{\text{uns}}$ is invertible, something is probably wrong with ΔG^{exp}."

7 Traceability: Tests Cases vs Requirements

Requirement	**Test**
R 1	Not covered in the Test Plan (GUI)
R 2	Test 13, Test 14, Test 15, Test 16, Test 17, Test 18, Test 19
R 3	Test 1
R 4	Not covered in the Test Plan (GUI)
R 5	Not covered in the Test Plan (GUI)
R 6	Test 20, Test 21, Test 22

Requirement	Test
R 7	Test 2, Test 3, Test 4
R 8	Not covered in the Test Plan (GUI)
R 9	Test 23, Test 24, Test 25
R 10	Test 5, Test 6, Test 7
R 11	Not covered in the Test Plan (GUI)
R 12	Test 26, Test 28, Test 27
R 13	Test 29, Test 30, Test 31, Test 32
R 14	Not covered in the Test Plan (GUI)
NR 1	Test 10, Test 11, Test 12
NR 2	Test 9
NR 3	Not covered in the Test Plan (GUI)
NR 4	Not covered in the Test Plan (GUI)
NR 5	Not covered in the Test Plan (Not a priority)
NR 6	Not covered in the Test Plan (Not a priority)
NR 7	Not covered in the Test Plan (Not a priority)
NR 8	Test 8
NR 9	Test 8

Table 2: **Traceability table: Test vs Requirements** [Capitalize Table —SS]

8 Prioritization

The resources available for the test phase of STEM Moiré GPA are relatively limited and all aspects of the test plan won't be covered. Therefore, the test tasks have been prioritized by focusing first on the system tests targeting the functional requirements, then the unit tests testing the functional requirements and finally the system tests assessing the non functional requirements. The details of the prioritization are described in the following table.

Priority	Requirement tested
P0	R 3 (Test 1), R 7 (Test 2, Test 3), R 10 (Test 5, Test 6)
P1	R 13 (Test 29, Test 30, Test 31, Test 32), R 12 (Test 26, Test 28, Test 27)
P2	R 7 (Test 4), R 10 (Test 7)
P3	R 2 (Test 13, Test 14, Test 15, Test 16, Test 17, Test 18, Test 19), R 6 (Test 20, Test 21, Test 22), R 9 (Test 23, Test 24, Test 25)
P4	NR 8 (Test 8) , NR 9 (Test 8)
P5	NR 1 (Test 10, Test 11, Test 12), NR 2 (Test 9)
P6	GUI and other lower priority tasks

Table 3: **Priority table** [Capitalize Table —SS]

[Great work! —SS]