XANDAR

Python More Image Than Memory (MITM) Viewer Software User Manual

TO-B251: FY23 GEOINT Active Sensors R&D Support

SUM-002

28 March 2025

Prepared by: Worden Barr and Isabel Wilson

1. Introduction

1.1 Purpose

The More Image Than Memory (MITM) application is a specialized software tool designed for displaying and performing simple processing operations on complex Synthetic Aperture Radar (SAR) data using the NGA Sensor Independent Complex Data (SICD) format.

1.2 Document Overview

This Software User Manual (SUM) is structured to provide users with all the necessary information to install and effectively use the MITM software. The manual is organized as follows:

1. **Introduction:** An overview of the software's purpose and design
2. **Installation Instructions:** Step-by-step guidance for setting up the software
3. **Graphical User Interface:** Detailed explanation of the interface and navigation
4. **Model View Controller:** Technical overview of the software architecture
5. **Conclusion:** Documentation summary
6. **Contact Information:** How to reach the developers

This manual assumes users have technical familiarity with SAR imagery and its applications but may not have extensive software development experience.

1.3 System Design

MITM is architected to efficiently handle large, complex SAR datasets while providing an intuitive interface for technical users. Several key design decisions inform the system architecture:

1.2.1 Model-View-Controller (MVC) Architecture

The software employs the MVC architectural pattern, offering several advantages:

- Clear separation of concerns, making the codebase more maintainable
- Modular design allowing independent development of components
- Simplified adaptation to changing requirements

1.2.2 PySide Graphics Library

MITM leverages the PySide (Qt for Python) framework for its graphical interface, which provides:

- Cross-platform compatibility (Windows, Linux)
- Native look and feel on each system
- Extensive widget library for specialized data displays
- Modern, customizable UI components

PySide was chosen over PyQt for its licensing and other advantages:

- LGPL license allows for both open-source and commercial use without requiring disclosure of application code
- No commercial licensing fees required for deployment
- Same Qt functionality and API as PyQt
- Growing community support, with more intuitive documentation

1.2.3 PyQtGraph Integration

MITM utilizes the PyQTGraph library for specialized image display capabilities:

- High-performance display of scientific/engineering image data
- Hardware-accelerated rendering for smooth interaction with large datasets
- Built-in support for complex image transformations and color mapping
- Interactive region-of-interest selection and analysis
- Seamless integration with the PySide framework
- Optimized for real-time updates and interactive visualization of SAR data

1.2.4 Memory Management Strategy

To handle SAR datasets that often exceed available system memory, MITM implements:

- Chunked data access patterns for efficient reading of large files
- Decimation strategy for image viewing that dynamically reduces resolution based on current zoom level and viewport size
- Intelligent disposal of out-of-view data chunks to minimize memory footprint

1.2.5 Concurrency Model

MITM employs a robust concurrent model based on Qt's QThread framework:

- Background processing for computationally intensive operations
- Responsive UI maintained during complex calculations
- Worker threads for data loading and processing operations
- Thread-safe communication via Qt's signals and slots mechanism

2. Installation Instructions

2.1 GitLab Download and Setup

If you are downloading MITM from a remote repository, follow the below steps to begin running MITM:

1. Navigate to the directory you would like to clone MITM into
2. $ git clone <"Clone with HTTPS/SSH" URL>
3. (Optional) Create a Python virtual environment (venv, conda, etc.) (Python 3.11 or greater)
4. Install the required packages by running: $ pip install –r requirements.txt
5. Run MITM: $ python main.py

2.2 Packaging MITM for Distribution

MITM is packaged using PyInstaller, which bundles Python, MITM, and all of its dependencies into a single executable which may then be distributed to other machines. The resulting executable is fully self-contained.

PyInstaller requires that MITM be packaged on the same system as the machines the resulting executable will be distributed to. For example, to run the executable on a Windows 10 machine, it must have been created on a Windows 10 machine, etc.

2.2.1 Packaging MITM as a Stand-Alone Application

How to package MITM as a stand-alone executable with PyInstaller:

1. pip install python-mitm (if it is not installed)
2. Navigate to the root of the MITM repository
3. $ cd standalone/pyinstaller
4. $ pyinstaller mitm.spec

How to run the resulting executable:

- $ ./dist/mitm.exe

2.2.2 Packaging MITM with Sub-Applications

Packaging MITM with sub-applications (e.g. RCS Tool) requires modifications to mitm.spec. These changes are outlined below. For best results, installing all sub-applications with pip is recommended. Variable names within mitm.spec are in bold.

1. Any non-python files the app relies on should be added to ***data_files*** along with the specific module that relies upon them.
2. The first argument to the ***Analysis*** object should be the path to the runner file (typically main.py). This path may be relative to mitm.spec or absolute.
3. Add the module name of the sub-application to the list passed to ***hiddenimports***. PyInstaller sometimes has difficulty finding modules and adding them here does not hurt.
4. (Optional) In the constructor for the ***EXE*** object, the ***name*** argument may be modified to change the name of the created executable.

3. Graphical User Interface (GUI)

3.1 Main Interface Overview

The MITM graphical user interface is designed to provide intuitive access to complex SAR data visualization capabilities.

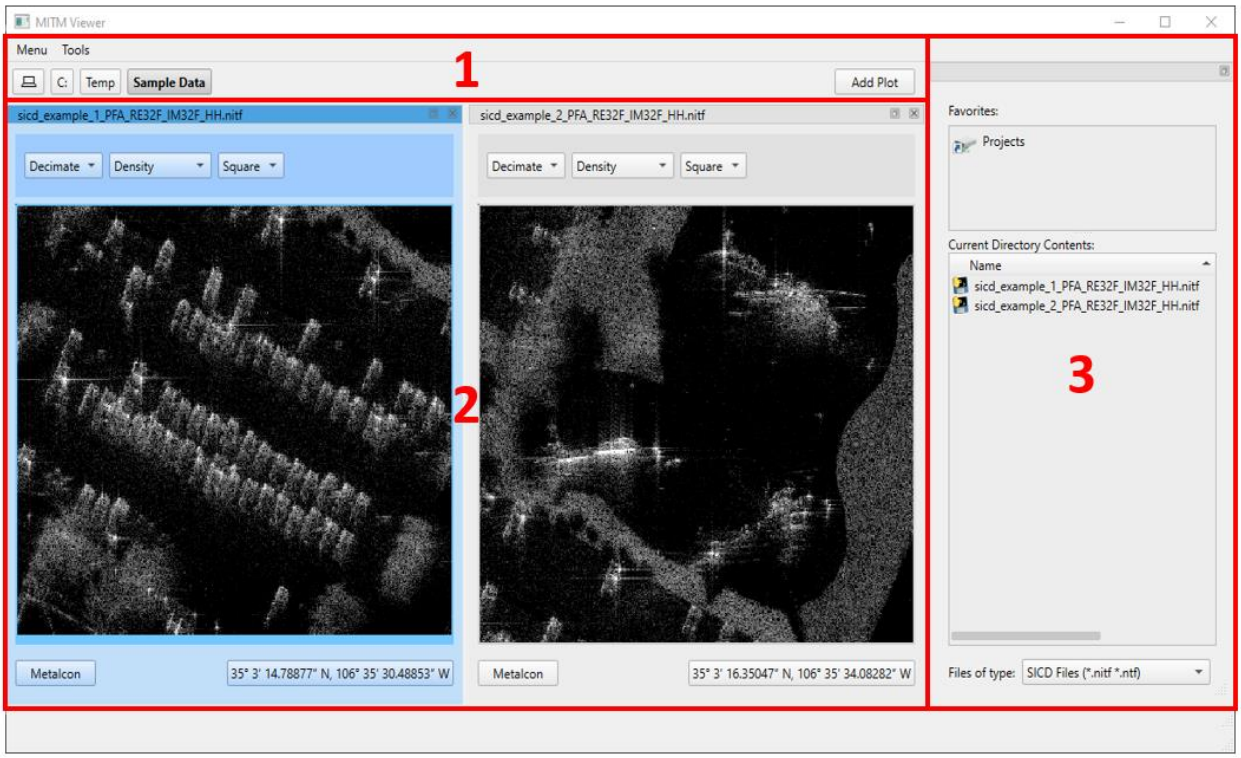

Figure 3a: MITM Main Interface

The main interface consists of three key components:

1. **Menu/Tool Bar:** Contains application commands organized in drop-down menus and directory navigation buttons
    a. Menu Dropdown: Contains options to toggle light/dark mode and open this SUM
    b. Tools Dropdown: If packaged with other SAR imagery tools, this dropdown will contain options to open those tools (e.g. RCS Tool)
    c. Directory Buttons: Clicking on any of the directory buttons will navigate the file browser to selected directory
    d. Add Plot: Adds an additional plot widget to the visualization area
2. **Visualization Windows:** Primary area where SAR imagery is displayed

3. **File Browser:** Simplified file and directory interface
    a. Favorites: Here the user can add favorite directories for quick navigation. Network storage location will also appear in this widget.
        i. To add directories to the Favorites widget, simply click and drag directories from the file browser into this area
    b. Current Directory Contents: This widget displays the contents of the current directory.
        i. *Double clicking directories* will navigate down the file structure.
        ii. *Double clicking SAR files* will open them up in a new plot.
        iii. *Clicking-and-dragging SAR files* into an open plot widget will display the image

The interface follows a modular design that allows users to customize their workspace by repositioning the plot widgets and the file browser according to their preference.

3.2 Plot Widget Interface Overview

MITM allows users to create multiple plot windows for viewing multiple images at once. Each plot widget provides several tools for re-presenting the selected SAR image.

Figure 3b: MITM Plot Widget Interface

1. **Plot Widget Header:** Displays the open file name. The user can click and drag in this region to move the widget elsewhere.
2. **Resampling Option Menu:** Adjusts the image decimation methods: (Decimate)
3. **Remapping Options Menu:** Allows choosing between several intensity remapping options: (Density, Brighter, Darker, High Contrast, Linear, Logarithmic, PEDF, NRL)
4. **Aspect Ratio Toggle Menu:** Toggles between square pixels and aspect ratio-adjusted pixels

5. **SAR Image Pane:** Area where SAR data is displayed
    a. Mouse wheel: Zoom in/out
    b. Click-and-drag: Pan
    c. Right click: Opens menu with options to copy cursor's geo coordinates or save the image to a file
6. **Metaicon Button**: Opens the MetaIcon window (Figure 3c)
7. **Geo Coordinates Button:** Displays real-time geolocation of the cursor. Clicking the button toggles between Degrees-Minutes-Seconds, Decimal Degrees, and Pixel Coordinates (X,Y)

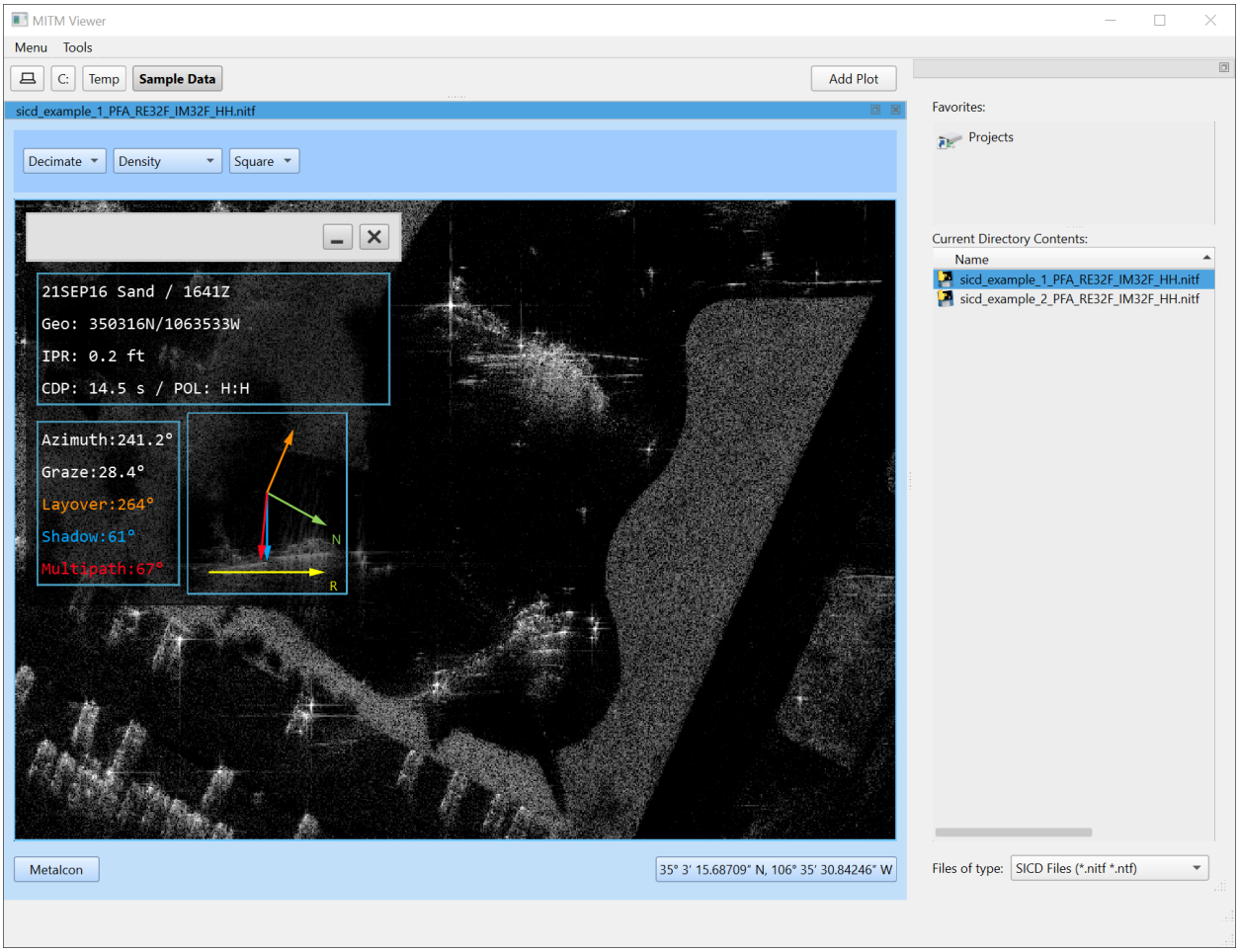

Figure 3c: MITM Image Display With Metaicon

3.3 Image Navigation Controls

MITM provides two methods for navigating through SAR images:

- **Zoom In/Out:** Mouse wheel up/down
- **Pan:** Click and Drag

Additional controls:

- Right clicking inside the image will open a menu with two options:
  - Copy Coordinates: Copies the current cursor coordinates (Figure 3d)
  - Export…: Exports the image to a file
- Clicking the ‘A’ in the bottom left of the image will re-center the image

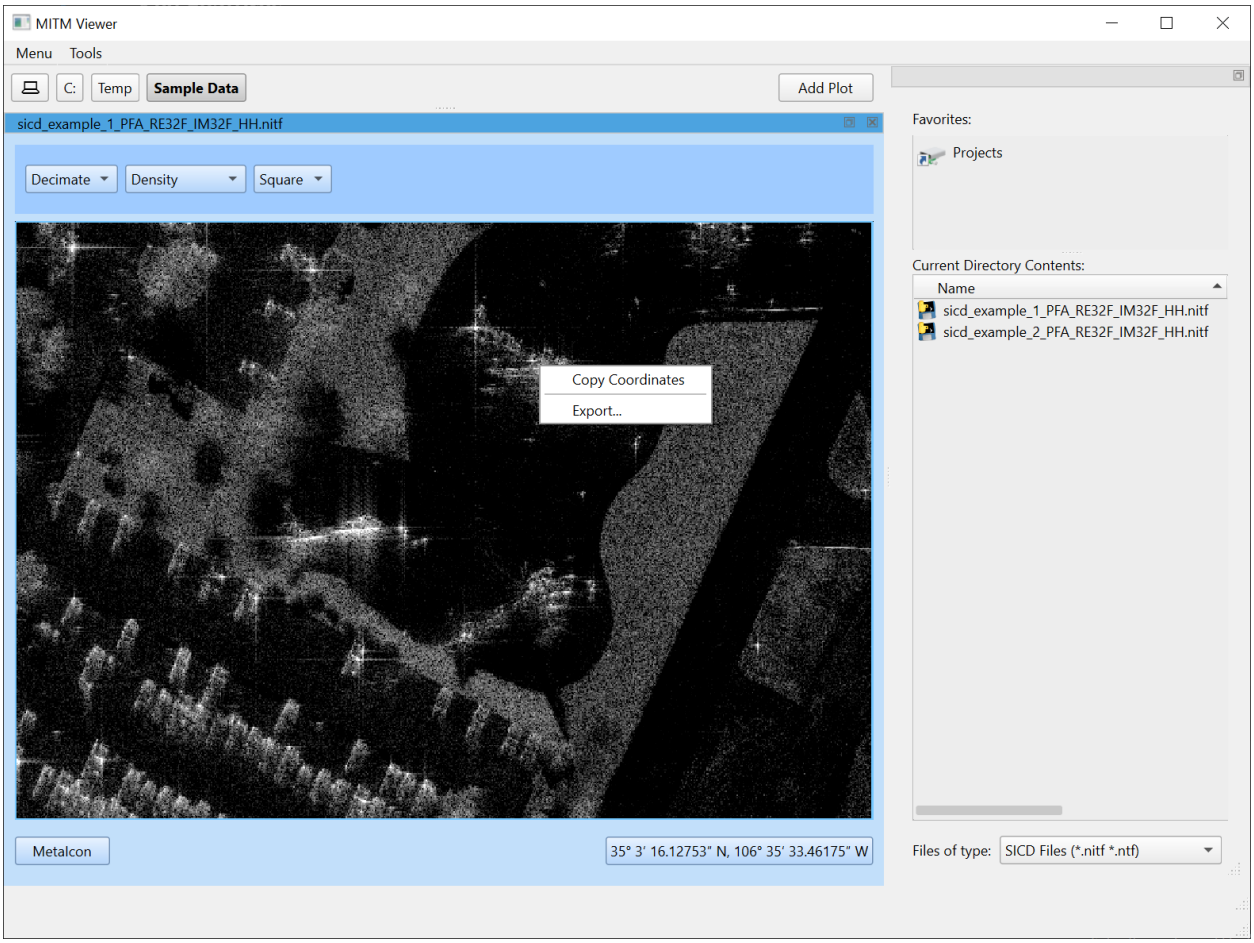

Figure 3d: Right-Click Menu

4. Model View Controller (MVC)

4.1 Architecture Overview

MITM is built using the Model-View-Controller (MVC) architectural pattern, which separates the application into three interconnected components. This separation enhances code maintainability and scalability.

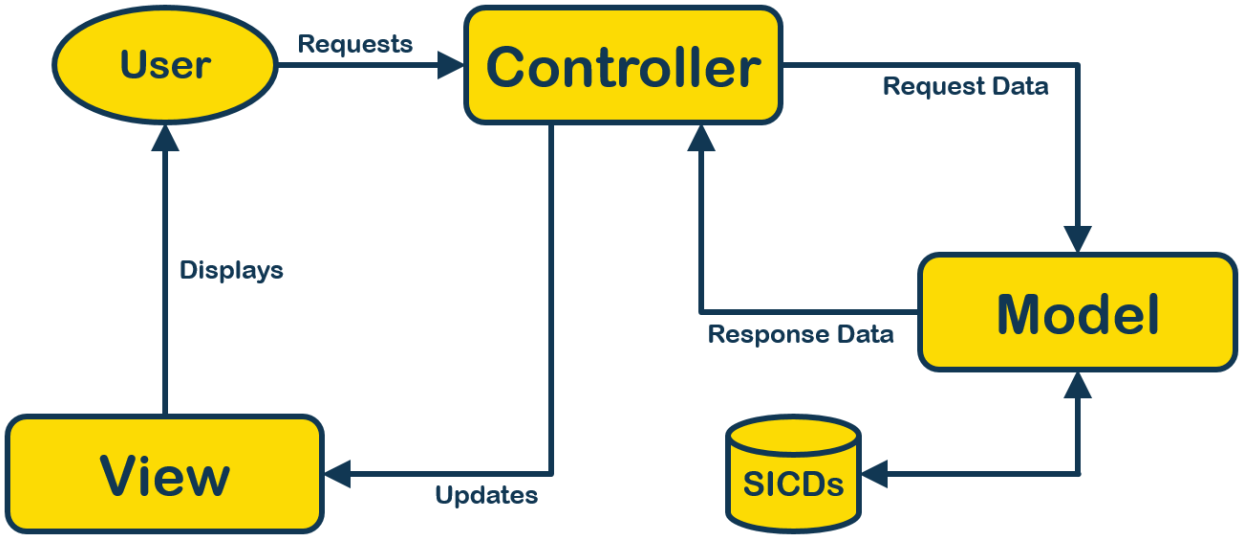

Figure 4a: MVC Architecture Communication Diagram

The three primary components of the MVC architecture are:

- Model: Manages data for the application
- View: Presents data to the user and handles user interface elements
- Controller: Processes incoming requests, manipulates data using the Model, and interacts with the View

4.2 Model

The Model is the application's dynamic data structure. It directly manages all of MITM's data sources and oversees the configuration state of the application.

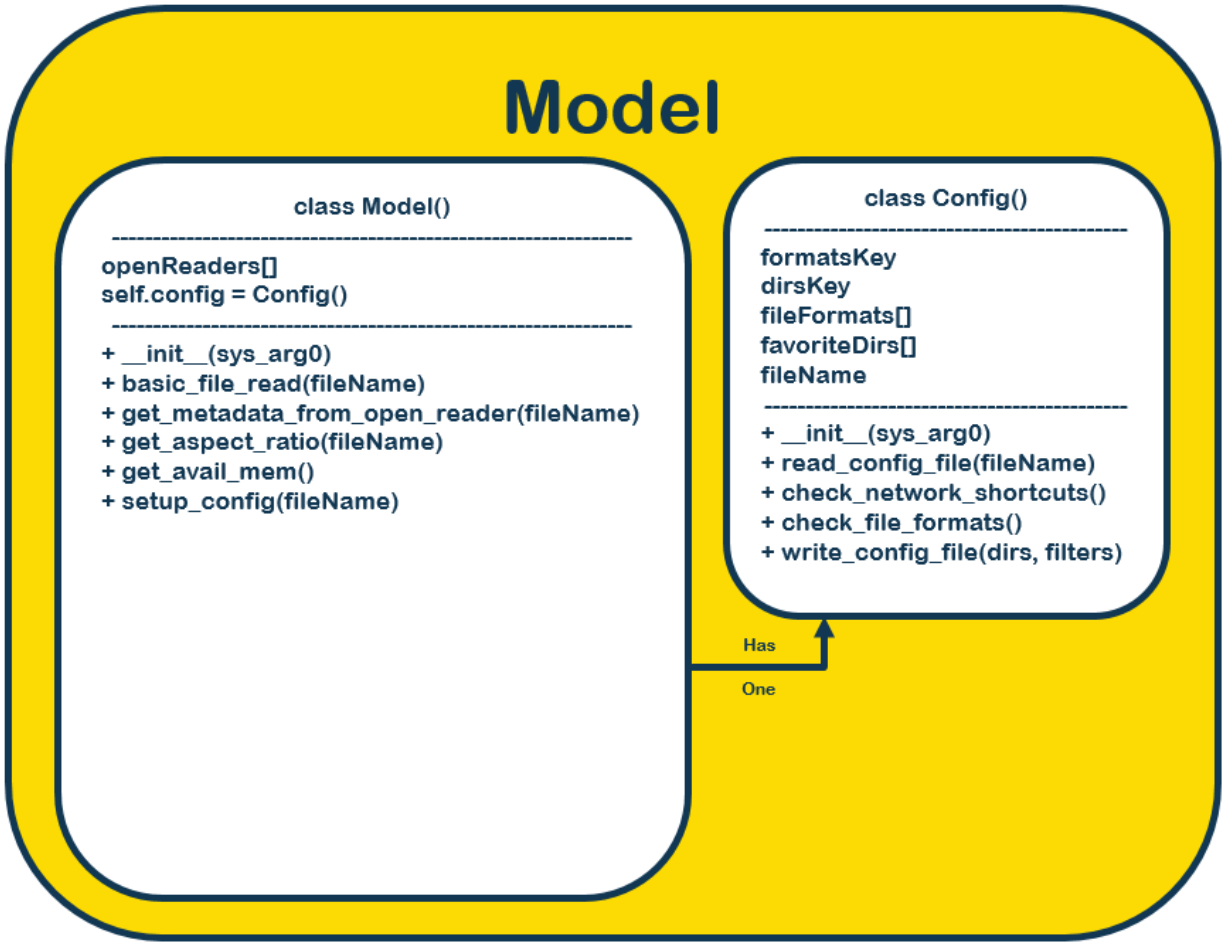

Figure 4b: Model Component Diagram

Key Responsibilities of the Model:

- **Data Loading:** Efficiently loads SICD-formatted SAR data
- **State Management:** Maintains a running state configuration for MITM
- **Metadata handling:** Extracts and provides access to SICD metadata

4.3 View

The View handles all user interface elements and data visualization in MITM.

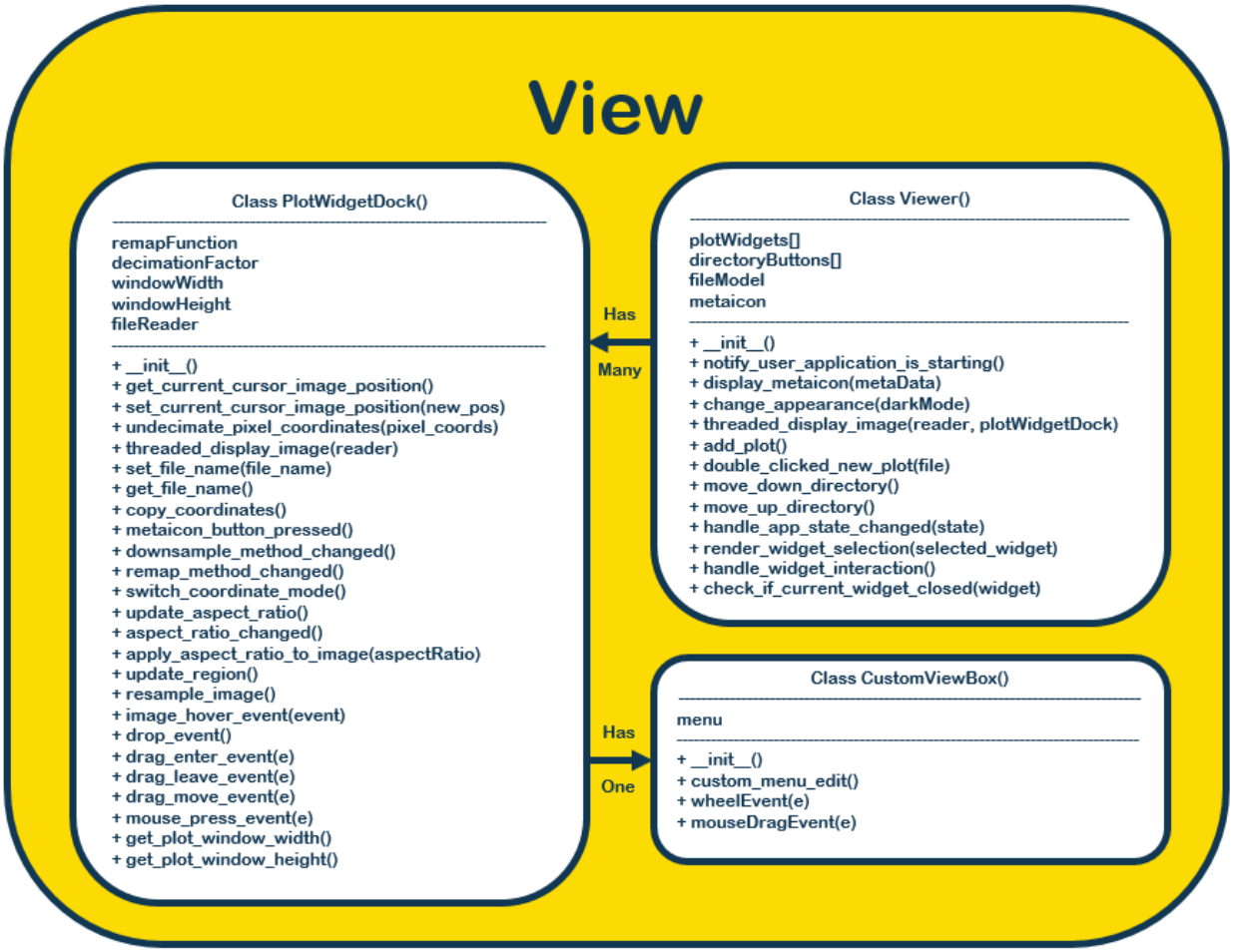

Figure 4c: View Component Diagram

Key Responsibilities of the View:

- **Interface Rendering:** Generates and manages all UI components
- **Data Visualization:** Renders SAR data according to selected visualization modes
- **User Input:** Captures mouse and keyboard interactions
- **Feedback Presentation:** Displays processing status and results
- **Adaptive Display:** Adjusts visualization based on available screen space and data characteristics

4.4 Controller

The Controller acts as an intermediary between the Model and View. It accepts input and converts it to commands for the Model or View.

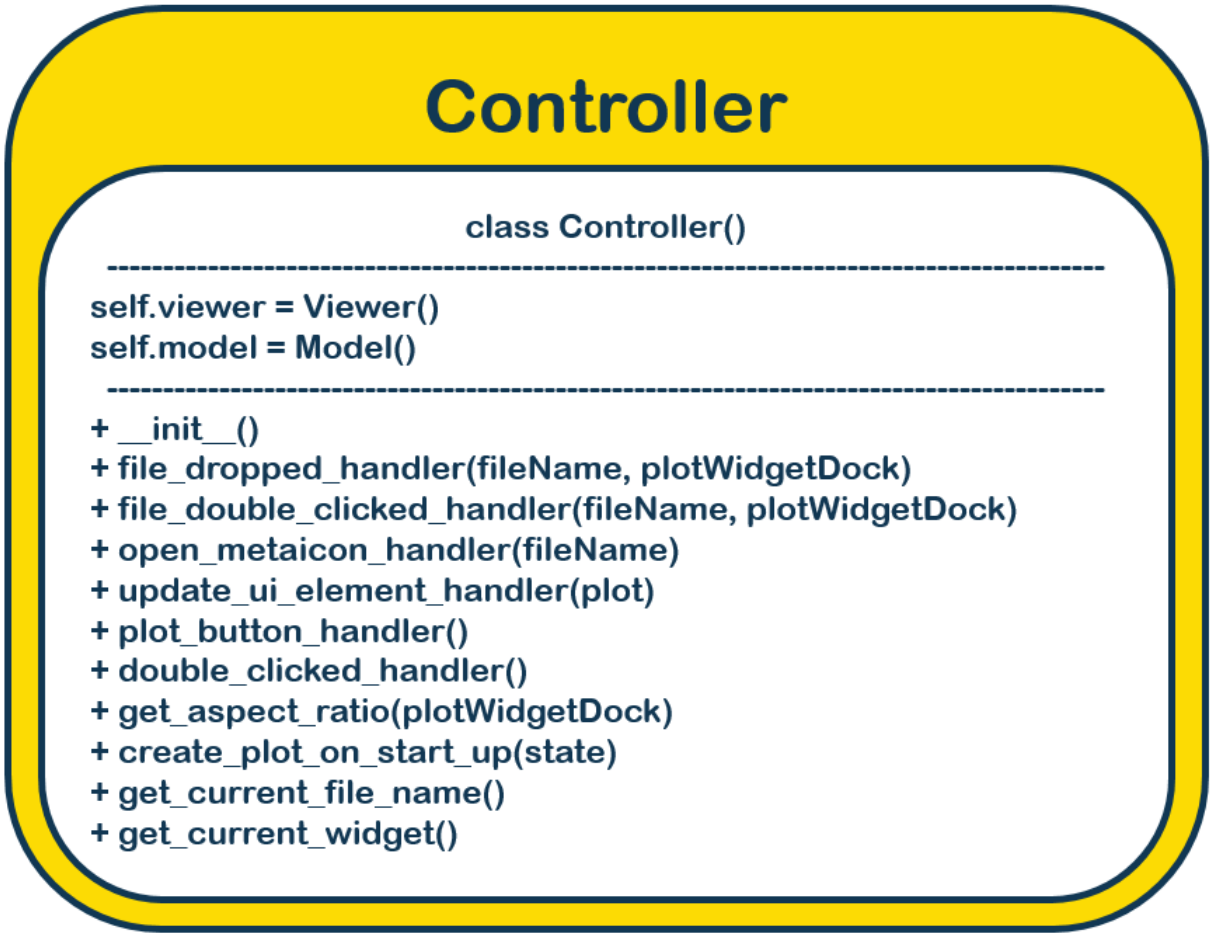

Figure 4d: Controller Component Diagram

Key Responsibilities of the Controller:

- **Command Processing:** Interprets user actions and triggers appropriate responses
- **Coordination:** Manages interactions between the Model and View components
- **State Transitions:** Handles application state changes
- **Event Management:** Processes and dispatches application events

5. Conclusion

The More Image Than Memory (MITM) Viewer provides practical capabilities for working with complex SAR data. By addressing the basic challenge of handling datasets larger than available memory, MITM allows users familiar with SAR imagery to view and analyze SICD-formatted data.

Key aspects of MITM include:

1. **Efficient Memory Usage**: Chunked data access and dynamic decimation strategies help manage large SAR datasets on systems with limited resources.
2. **Practical User Interface**: The modular GUI with customizable plot widgets lets users organize their workspace as needed.
3. **Solid Technical Design**: The Model-View-Controller architecture paired with PySide and PyQtGraph libraries provides a stable foundation.
4. **Multiple Visualization Options**: Different resampling and remapping options help users adjust their view of SAR data as needed.

SAR datasets continue to grow in size and complexity, making tools that can handle large files increasingly necessary. MITM aims to provide a straightforward solution for this challenge.

For questions or additional information, please refer to the contact details provided in section 6.

6. Contact Information

Name	Role	Email	Phone
Richard Borth	Government POC	richard.borth.1@us.af.mil	937-522-6251
Worden Barr	Developer	barr-worden@n-ask.com	
Isabel Wilson	Developer	wilson-isabel@n-ask.com	
Paul Harmer	Team Lead	paul.harmer@us.kbr.com	937-797-6762