**Task Progress Update Report**

**Name**: LIM SHI KAI (Sky)  
**Update Date**: 14-11-2024

# **Overview of Assigned Tasks**

**Task 1: Line Information Box for Detect and Remove Line function**

* **Objective:** Display information about detected and removed lines to facilitate tracking of total lines detected and removed.
* **Current Status: Completed**
* **Details:** 
  + The detected lines display includes coordinates, width, and line type (e.g., isolated or within-object lines).
  + The removed lines display includes details such as the removal method, line types, initial quantity of lines, removed line data, remaining line data, and a summary of remaining lines.
* **A screenshot of a computer

  Description automatically generatedImage:**

Figure 1.1 The line information box when detecting the lines

A screenshot of a computer

Description automatically generated

Figure 1.2 Line information box after removed line

**Task 2: Remove Line Method**

* **Objective:** Enable removal of lines with larger widths through a direct stitch method.
* **Current Status:** Completed
* **Details:** 
  + The function now supports two removal methods:
    - **Use Neighbor Values:** Detects line width automatically and dynamically sets the SEARCH\_RADIUS to 1.5 times the actual line width, ensuring that neighboring values are replaced efficiently.
    - **Direct Stitch:** Removes the line directly, decreasing the image size if it exceeds a specific width (recommended for wide images).
  + Users can select line types, removal methods, and specific lines to process (applicable for in-object lines):
    - **For Neighbor Values:** Users can select individual lines or use the “Select All Lines” option.
    - **For Direct Stitch:** Only a single line can be processed at once to avoid affecting the output quality.

A screenshot of a computer program

Description automatically generated

Figure 1.3 Control Panel for Removing Line

**Task 3: Zoom Mode Updating**

* **Objective:** Allow users to zoom in or out, fix the image at a specific zoom level, and view processing results without scrolling.
* **Current Status:** Completed
* **Details:** 
  + Implemented zoom functionality with zoom-in, zoom-out, and reset features. When zoom mode is active, other process buttons are disabled to prevent conflicts.
  + Once zoom mode is fixed, users can proceed with processing at the desired zoom level.
  + Visual cues for zoom status: “Zoom” button changes to blue when activated and red when fixed; “Fix Zoom” button also changes to blue when fixed. These indicators help users easily identify the current zoom status.
* **Image:**

A screenshot of a computer

Description automatically generated

Figure 1.4 Zoom Mode and Detect Line function

**Task 4: Restructure Detect and Remove line into double 2D Pointer method**

* **Objective:** Enhance the detect and remove line function by integrating a double 2D pointer structure and incorporating it into the existing system.
* **Current Status:** In Progress
* **Details:** 
  + The function has been restructured into ImageData, a struct for storing double 2D pointers and integer values for rows and columns.
  + Successfully integrated the current detect and remove functionality.
  + Output remains consistent with the previous implementation.
  + Dark lines are stored using a vector; confirmation on this storage approach is pending.

# **Roadblocks/Challenges**

# updateImageDisplay() Reconstruction: Required to modify updateImageDisplay() to accommodate the detected line functionality in a new method, as the detected line label size was affected.

# Pointer Linkage Issues: Encountered frequent crashes due to pointer linkage with the control panel.

# Zoom Button Color Change: Initial challenges with color change for zoom buttons were resolved by implementing a logic-based approach to detect button names, enabling accurate color changes.

# Control Panel Testing: Multiple tests are necessary to ensure smooth functionality of the Control Panel for removing lines, along with verifying the processing logic to maintain image integrity.

# **Conclusion**

* Successfully implemented enhancements for line detection and removal, including a detailed line information display.
* Refined line removal methods to support both neighbor-value replacement and direct stitch, expanding versatility for different line types and widths.
* Updated zoom mode to allow users to fix the zoom level during processing, with clear visual indicators for zoom status.
* Overcame challenges related to:
  + **updateImageDisplay()** adjustments for new line detection methods,
  + Pointer linkage issues causing crashes,
  + Zoom button color change inconsistencies, and
  + Control panel testing for line removal functions.
* Remaining tasks include finalizing the 2D pointer-based approach for line detection and removal and conducting additional tests to ensure performance and user-friendliness.