

## Objectives

1. Implement an enhanced "Interlace" function to streamline merging, eliminating the need for an extra merge button.
  2. Update the control panel to support the new interlace method with merge options.
  3. Refine the Threshold CLAHE function for CPU and GPU modes to improve output quality by reducing noise and enhancing image detail in dark regions.
  4. Manage code organization for easier interlace function management and debugging.
- 

## Activities

1. Implemented a refined interlace function that now merges automatically without requiring an additional button press. The new function operates according to the same interlace rules as before but enhances workflow efficiency.
2. Updated the control panel by adding an "Enhanced Interlace" option in the Pre-Processing section, allowing users to choose the merge method and start interlacing seamlessly.
3. Verified that the revert logic works correctly with the enhanced interlace function, ensuring smooth backtracking if needed.
4. Moved interlace-related functions into separate files (`interlace.h` and `interlace.cpp`) to simplify code management.
5. Reviewed Threshold CLAHE outputs and observed noise in both CPU and GPU versions, prompting parameter adjustments, such as modifying the threshold multiplier and applying Gaussian factors. Reverted temporary adjustments due to suboptimal results.
6. Tested dynamic CLAHE adjustments for clip limits and tile sizes based on dark regions, allowing finer control over the threshold CLAHE effect.
7. Created a new function to identify regions that require CLAHE based on threshold values for better detail enhancement.
8. Identified differences between threshold CLAHE on GPU and CPU modes, noting the GPU version's more complex processing algorithm. Temporarily switched GPU threshold CLAHE code to a CPU-based approach to unify algorithm and parameter consistency across both modes.

9. Made additional adjustments to threshold CLAHE, attempting to balance detailed enhancement and noise reduction, but encountered OpenCV errors, leading to a temporary pause on further changes for stability.
  10. Ensured that threshold CLAHE only processes pixels below the threshold, storing them in a vector for processing before restoring them to their original values in the final image.
  11. Noticed the previous threshold CLAHE method relied on a mask, mixing functions incorrectly. Re-implemented the vector-based approach, with plans for further processing to achieve optimal output.
- 

## Achievements

1. Successfully implemented an "Enhanced Interlace" function that merges automatically, improving usability by removing extra steps.
  2. Updated the control panel to align with the new interlace process, enhancing user interaction.
  3. Unified the threshold CLAHE algorithm across GPU and CPU modes to facilitate consistent output and easier troubleshooting.
  4. Reorganized the interlace code into separate files, making the codebase more modular and manageable.
- 

## Problems & Solutions

- **Problem:** Threshold CLAHE outputs showed significant noise and blurriness.
  - **Solution:** Attempted parameter adjustments and dynamic CLAHE tuning for dark regions. Implemented a vector-based approach for threshold CLAHE to replace the incorrect mask-based method. Further refinement is planned after investigating OpenCV errors.
- **Problem:** Differences between GPU and CPU processing algorithms caused inconsistencies in threshold CLAHE output.
  - **Solution:** Rewrote the GPU threshold CLAHE using the CPU algorithm to ensure matching outputs, simplifying debugging and parameter adjustments.
- **Problem:** OpenCV errors prevented detailed testing of the revised threshold CLAHE output.

- **Solution:** Retained the current code without additional adjustments to enable error isolation and troubleshooting in the next session.
- **Problem:** The initial threshold CLAHE function applied a mask incorrectly, affecting output quality.
  - **Solution:** Transitioned to a vector-based method for identifying and processing pixels below the threshold, with further steps planned to improve the final output.