

Objectives

1. Refine and enhance the `detectDarkLines` function to identify and characterize dark lines in the image based on various thresholds and conditions.
 2. Resolve issues with visualizing detected lines and updating their states after removal.
 3. Improve line detection accuracy for vertical and horizontal orientations.
 4. Integrate a new algorithm to distinguish lines near objects from those in isolated regions, ensuring that only appropriate lines are removed.
 5. Implement multi-threading for functions such as Gamma, Sharpen, and Contrast, enhancing both overall and regional processing.
-

Activities

1. Reimplementation of `detectDarkLines` Function:

- Redesigned `detectDarkLines` to accept `threshold`, `dark threshold`, and `min length` as parameters.
- Added helper functions:
 - `isInBrightRegion`: Checks if a point meets the light threshold.
 - `refineDarkLineDetection`: Merges and smooths detected dark lines.
 - `calculateLineThickness`: Calculates and records the width of each detected line.
 - `findBrightRegions`: Identifies bright regions within the image.

2. Visual State Management and Line Visualization:

- Addressed visualization issues, including clearing red highlight lines after removal.
- Ensured all relevant functions call `clearDetectedLine` to reset visual indicators.

3. Enhancing Line Detection Orientation:

- Enabled detection of both vertical and horizontal lines through `x-axis` and `y-axis` traversal.
- Improved functionality to detect lines regardless of black or bright region characteristics.

4. Object-Based Line Removal Algorithm:

- Added `inObject` flag within the `BlackLine` struct to mark lines associated with objects.

- Set a separate threshold (`OBJECT_THRESHOLD`) for dark objects, allowing selective line removal.
- Refined detection sensitivity using a neighboring-pixel threshold for accurate in-object vs. isolated line identification.
- Adjusted `WHITE_THRESHOLD` to 55000, improving detection accuracy for different input files.

5. Thread Processing and Debugging:

- Implemented multi-threading for Gamma, Sharpen, and Contrast functions.
- Added debugging outputs in the console to show processing progress for `detectDarkLines` and line removal functions.
- Updated the info bar to display detected line coordinates and weight, streamlining information display and image view.

Achievements

1. Completed reimplementation of `detectDarkLines` with enhanced functionality and helper methods for line detection and refinement.
2. Resolved visual state management issues for detecting, highlighting, and removing lines.
3. Achieved robust line detection across vertical and horizontal orientations.
4. Finalized an object-based detection algorithm, allowing selective removal of lines outside of objects with improved threshold tuning.
5. Successfully integrated thread processing for key functions, enhancing processing efficiency and feedback in the debug console.

Problems and Solutions

Problem 1: Visualization issues with highlighted lines persisted after line removal, with some vertical lines not being cleared properly.

Solution 1: Revisited the state management flow and ensured that `clearDetectedLine` was called by all relevant functions to reset highlighted lines post-removal.

Problem 2: Only vertical lines were initially detected, limiting the function's utility.

Solution 2: Adjusted detection to process lines along both `x-axis` and `y-axis`, enabling full vertical and horizontal line detection.

Problem 3: Overly sensitive object detection led to incorrect line removal in object-adjacent regions.

Solution 3: Added `inObject` flag to track line proximity to objects and refined detection using a neighboring-pixel check with `WHITE_THRESHOLD` set to 55000, achieving accurate isolated vs. object-based line removal.

Problem 4: Threading in new functions required debugging for synchronized processing outputs.

Solution 4: Added detailed debugging in the console to track thread processing across Gamma, Sharpen, and Contrast functions, ensuring reliable processing feedback.