Problem Statement:

In this task, we aim to process image data to track spherical markers that appear as bright blobs in camera images. The provided data consists of lines of bright pixels extracted from image rows, sorted by their Y and X coordinates. Each line represents a segment of a marker blob. The objective is to assign a unique blob index to each line such that lines belonging to the same blob share the same index.

A blob is defined as a set of touching lines, meaning two lines belong to the same blob if they are on consecutive rows and share at least one common pixel in the column range. The algorithm should efficiently cluster lines into blobs and assign them sequentially indexed labels.

Challenges:

1. Handling large images with numerous blobs efficiently.
2. Correctly merging touching lines while ensuring that disconnected groups receive separate indices.
3. Managing bridge cases where multiple disconnected lines in a row are connected by a line in the row below.

Solution Approach: The solution is inspired by agglomerative clustering, where small clusters (lines) are successively merged into larger clusters (blobs) based on their touching condition. The key steps include:

1. **Initialization**: Create a mapping of blob indices and initialize all lines with a default blob index.
2. **Iterative Processing**:
    - Traverse the sorted line list and check for touching conditions with previous rows.
    - If a line touches any existing blob, merge them.
    - If a line touches multiple blobs, merge all affected blobs into one.
    - If a line does not touch any existing blobs, create a new blob entry.
3. **Final Mapping**: Assign the computed blob indices to each line.

The implementation makes use of efficient data structures such as hash maps and vector lists to optimize lookup and merging operations.

Performance Considerations:

- The algorithm minimizes redundant lookups by leveraging an ordered structure.
- Efficiently merges blobs by only updating relevant entries rather than iterating through all previous lines.
- Handles edge cases such as bridges by intelligently merging blob indices when a connection is found.

This approach ensures correct blob detection with an optimized processing time suitable for large-scale image data.

#