CSL7360 - Computer Vision

Project Report

Group Member:

Harsh Kumar (B20AI011)

Saurav Kumar (B20EE081)

Prashant Kumar (B20ES013)

Utkarsh Gupta (B20ES009)

Panoramic Image Stitching

Problem Statement:

The project aims to develop a Computer-Vision based automatic panoramic image stitching. Panoramic images provide a wider field of view compared to standard images, making them useful for capturing landscapes, architectural structures, and other wide-scene environments. However, manually stitching multiple images together to create a panoramic view can be time-consuming and error-prone. Therefore, the project addresses this issue by automating the process of image stitching using computer vision techniques.This concept is extrapolated to solve the problem of **video stitching** as well.

Description:

Panoramic image stitching is the process of combining multiple images with overlapping fields of view to create a single wide-angle or panoramic image. This technique is commonly used in photography to capture landscapes, architecture, and other scenes that cannot be fully captured in a single frame.

Panoramic video stitching is the process of combining multiple videos frame by frame with overlapping fields of view to create a single wide-angle or panoramic video.

Methodology:

Our methodology works in three phases:

1) Implemented image stitching from scratch to make horizontal panorama and then done analysis of the stitching and then modified the algorithm to stitch the image vertically as well (vertical panorama).
2) Implemented image stitching using a stitcher library with various feature detection algorithms like SIFT,ORB, BRISK and then did deep analysis of how the algorithm works.

3) Then, I implemented a video stitching algorithm using the stitching algorithm used in the above two steps and finally was able to do video stitching with good results.

Methodology 1(Scratch Implementation):

1. **Image Preprocessing:**
   - Prior to stitching, the input images undergo preprocessing to ensure better alignment and blending.
   - This preprocessing may include adjusting the exposure, white balance, and removing lens distortion using techniques like barrel or pincushion distortion correction and warp perspective.
   - Preprocessing aims to enhance the visual quality of individual images and improve the overall stitching process.
2. **Feature Detection and Matching:**
   - The SIFT (Scale-Invariant Feature Transform) algorithm is employed for detecting distinctive key features and descriptors within each image.
   - SIFT detects keypoints based on local gradients and generates descriptors to represent the local image regions around these key points.
   - Key feature points and descriptors are extracted from all input images.
3. **Homography Estimation:**
   - Once keypoints and descriptors are obtained, the next step involves matching corresponding key points between pairs of adjacent images.
   - The matching process aims to find the best correspondences between keypoints in different images.

- The **RANSAC (Random Sample Consensus)** algorithm is utilized to estimate the homography matrix, which represents the geometric transformation between image coordinates.
- RANSAC robustly estimates the homography by iteratively selecting subsets of keypoint correspondences and fitting a homography model to these subsets.

4. **Image Alignment:**
    - Using the computed homography matrix, the images are aligned to correct for differences in perspective, rotation, and scale.
    - Each image is transformed according to its homography matrix to ensure proper alignment with the reference image.
    - Image alignment ensures that overlapping regions between adjacent images match seamlessly, resulting in a visually coherent panorama.
5. **Blending:**
    - After alignment, the images are blended together to create a smooth transition between overlapping regions.
    - Various blending techniques such as feathering, gradient blending, or multi-band blending may be employed to achieve seamless transitions.
    - Blending ensures that there are no visible seams or artifacts at the boundaries of overlapping regions, producing a visually pleasing panoramic image.
6. **Vertical Stitching:**
    - In addition to horizontal stitching, the methodology includes functionality for vertical stitching.
    - Vertical stitching is applicable when images are stacked vertically to capture a wider field of view.
    - The same feature detection, matching, homography estimation, alignment, and blending steps are applied to handle vertically

stacked images, resulting in a panoramic view that covers both horizontal and vertical perspectives.

Methodology 2 (Using Library) :

1. **Initialization and Input Preparation:**
   - The process begins by importing the necessary libraries and initializing the `Stitcher` object provided by the stitching library.
   - Input images are provided as a list of file paths, representing the images to be stitched together.
2. **Image Stitching:**
   - The `stitch` method of the `Stitcher` object is invoked with the list of input image paths.
   - The Stitcher library internally performs feature detection, matching, homography estimation, image alignment, and blending to create a stitched panorama.
   - The output panorama is a single stitched image representing the seamless combination of input images.
3. **Visualization and Analysis:**
   - The stitched panorama is displayed using the `plot_image` function, providing a visual representation of the stitching result.
4. **Image Resizing and Resolution Analysis:**
   - The input images are resized to different resolutions (medium, low, and final) using the `Images` class provided by the stitching library.
   - Resized images are plotted to visualize the effect of resolution reduction on image quality and content.
5. **Feature Detection and Matching:**
   - Feature detection and matching are performed using the `FeatureDetector` and `FeatureMatcher` classes, respectively.
   - Key features are detected in the input images, and matches between features are identified to establish correspondences between images.

- Confidence matrices and match visualization are generated to assess the quality and reliability of feature matches.

6. **Subset Selection and Image Cropping:**
    - The `Subsetter` class is utilized to select a subset of images based on feature matching quality.
    - Additionally, the `Cropper` class is employed to crop images and masks based on the largest interior rectangle (LIR) to remove unwanted areas and focus on the relevant content.
7. **Image Warping and Timelapse Generation:**
    - The `Warper` class is utilized to warp images based on estimated camera parameters and create panoramic views.
    - Timelapse generation is performed using the `Timelapser` class to sequentially process warped images and create a continuous panoramic video.
8. **Seam Finding and Exposure Error Compensation:**
    - Seam finding is executed using the `SeamFinder` class to identify seams between adjacent images and minimize visible artifacts.
    - Exposure error compensation is applied using the `ExposureErrorCompensator` class to adjust image exposure and brightness for seamless blending.
9. **Image Blending and Final Panorama Generation:**
    - Image blending is performed using the `Blender` class to blend warped images and masks, ensuring smooth transitions between images.
    - The final panorama is generated by blending the compensated images, resulting in a high-quality stitched panorama.
10. **Visualization of Seam Masks and Analysis:**
    - Seam masks generated during the blending process are visualized to assess the effectiveness of seam finding and blending techniques.
    - Seam lines and polygons are drawn on the final panorama to visualize the seam areas and ensure seamless integration of images.

Methodology 3: Video Stitching

1. **Video to Frames Conversion:**
    - The process begins by converting input videos into individual frames, which facilitates frame-level processing.
    - The convert_video_to_frames function reads each frame from the input video files and saves them as separate image files (e.g., JPEG format) in designated output folders.
    - Each frame is extracted using the cv2.VideoCapture module in OpenCV, and the frames are sequentially numbered for easy referencing.
2. **Frame Stitching:**
    - Once the frames are extracted, they are divided into two sets corresponding to the two input videos.
    - For each pair of frames from the two sets, the frames are stitched together horizontally using the stitch_video_frames function.
    - The stitch_video_frames function utilizes the previously implemented Panaroma class(from the scratch implementation) or stitcher function from library to perform image stitching, similar to the horizontal image stitching process described earlier.
    - The resulting stitched frames represent panoramic views combining frames from both input videos, creating a continuous scene.
3. **Frames to Video Compilation:**
    - After stitching, the stitched frames are compiled back into a video format using the compile_frames_to_video function.

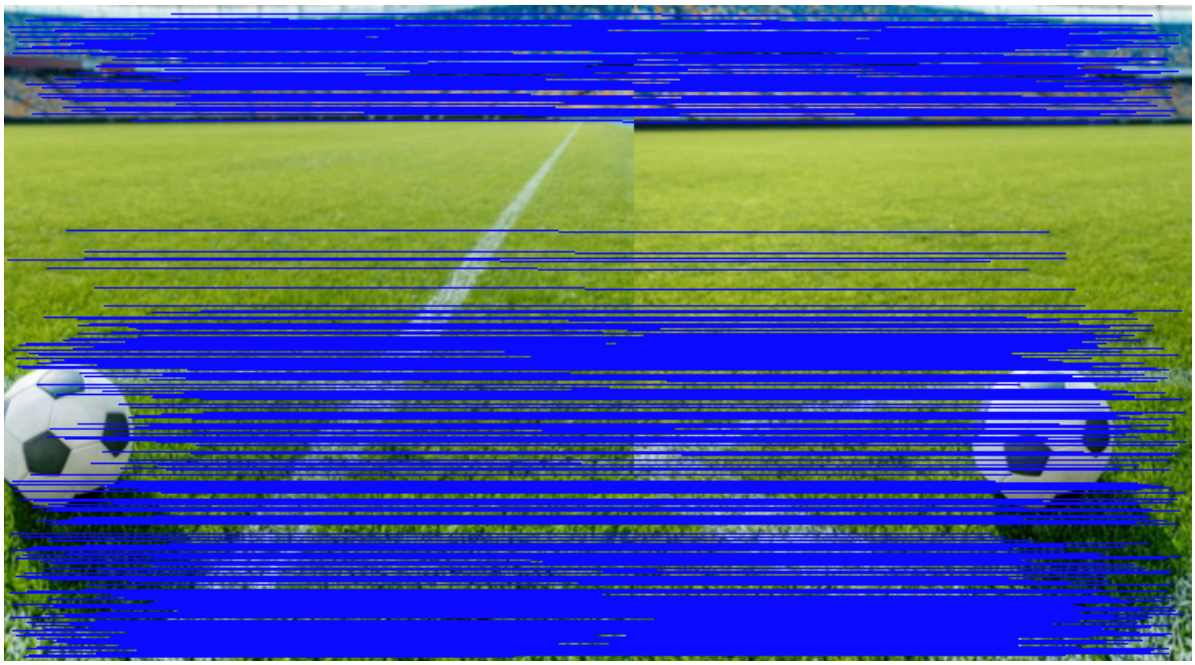
- The function reads the stitched frames from the output folder, sorts them based on their filenames, and writes them sequentially into a new video file.
- OpenCV's VideoWriter class is used to create the output video file, specifying parameters such as video codec, frame rate, and dimensions based on the first frame's properties.
- The final output is a single video file containing the stitched panoramic views, ready for playback or further processing.

Results:

Horizontal Image Stitching:

Input Image:

Featured matched points:

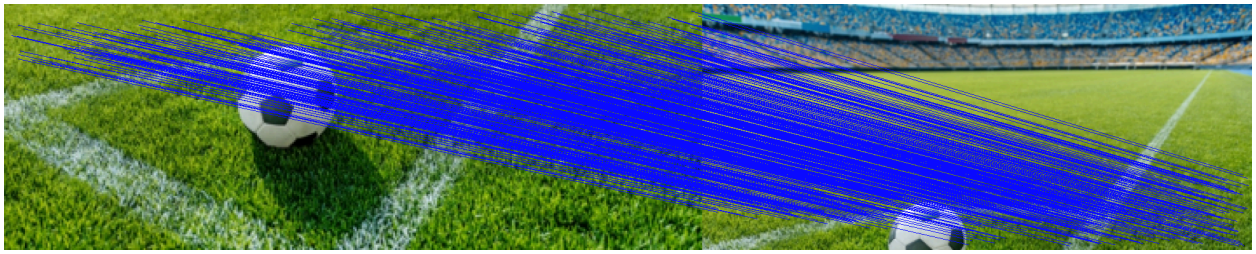


Final image after stitching:

Vertical Image Stitching:

Input Image:

Featured matched points:



Final image after stitching:

Result for Image stitching using stitcher:

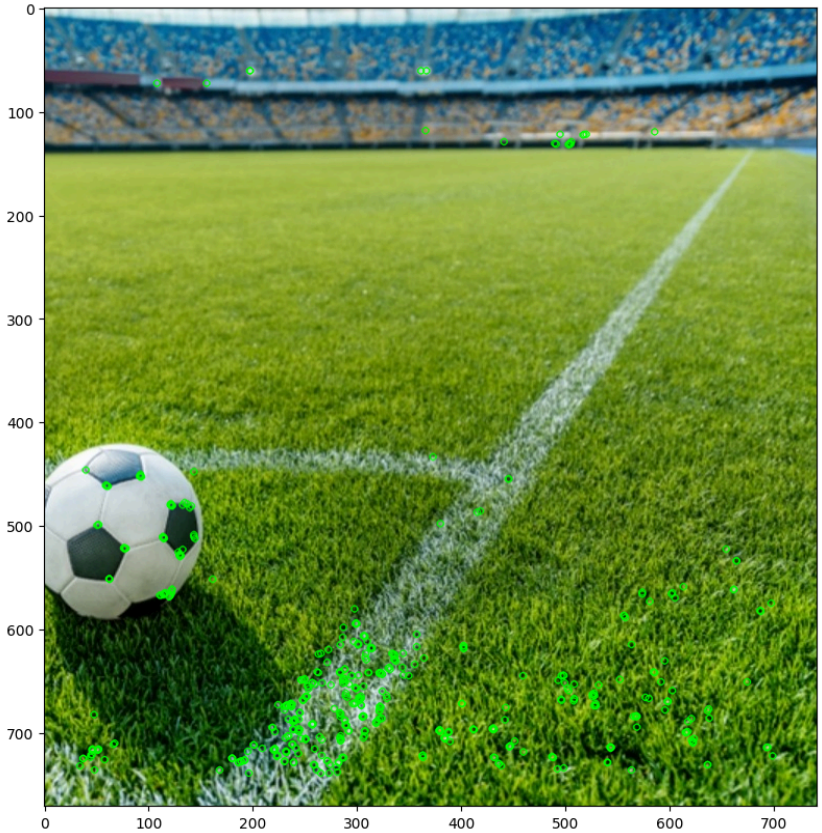
Resolutions of image for stitching:

Medium Resolution

Low Resolution

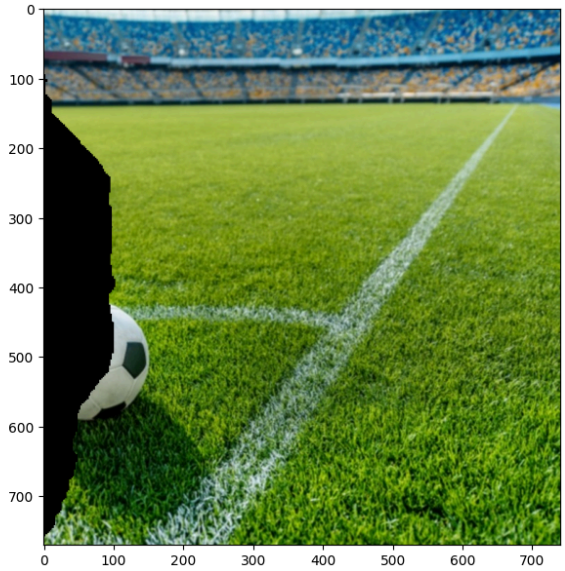
Final Resolution

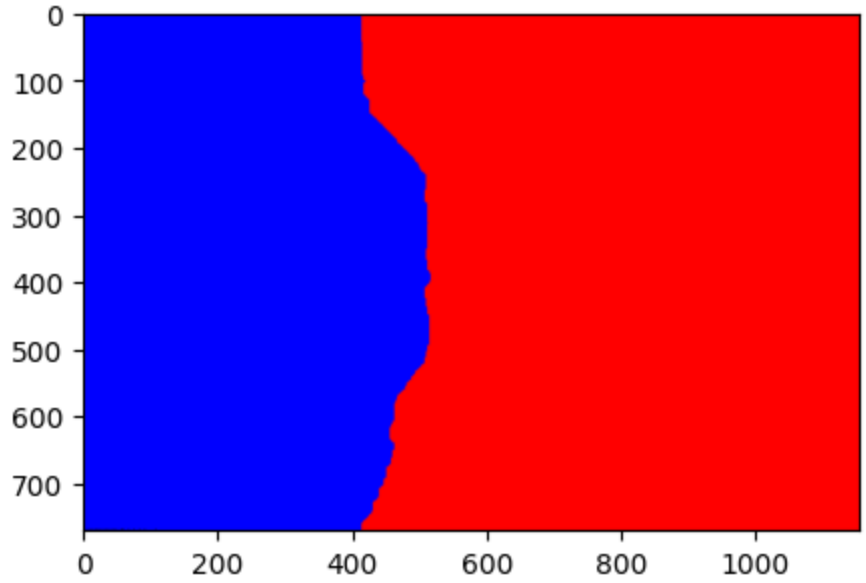
Key point center:



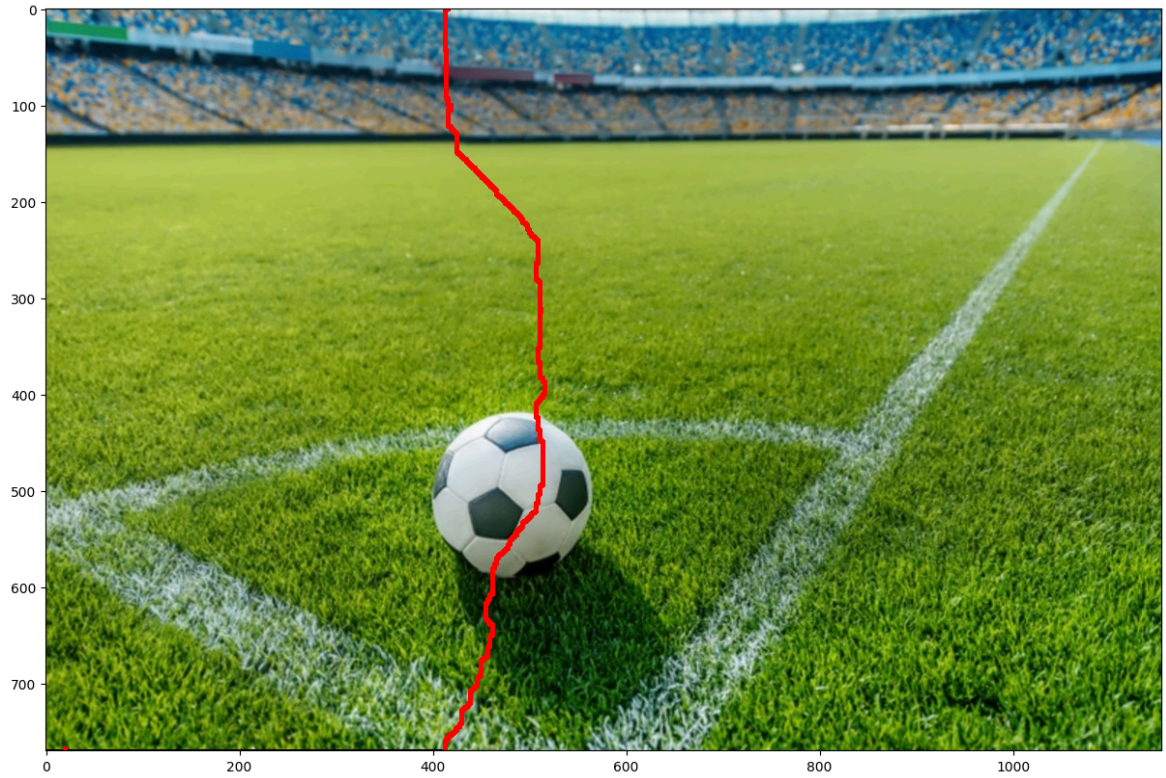
Feature matched point:

Seam finder:





Seam with line:

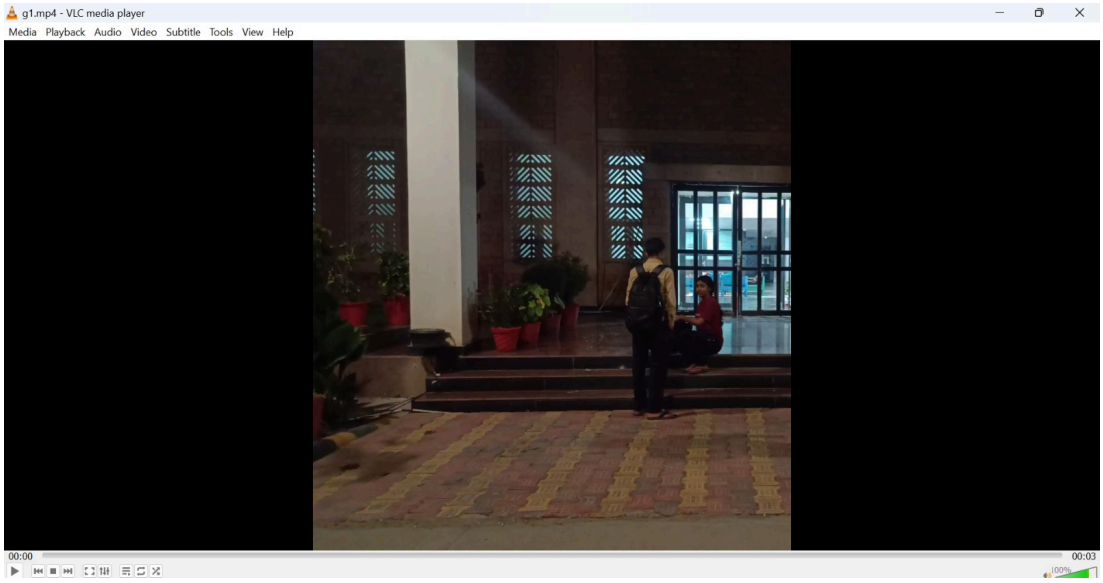


Seam with polygon:

Final image after stitching:

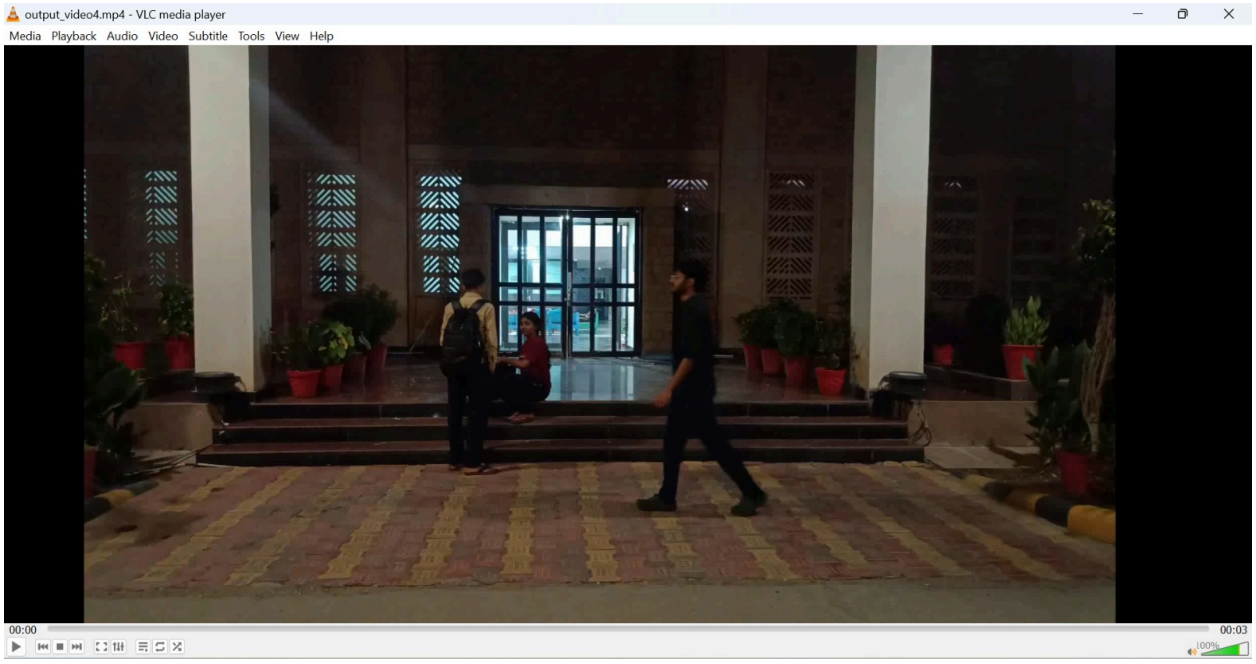
Video Stitching:

Input left video:



Input Right video:

Output stitched video:



Observation and calculation:

The project explored various techniques for panoramic image stitching, ranging from manual approaches to automated methods using libraries and custom stitching algorithms. Through experimentation and implementation, the following conclusions can be drawn:

- Automated stitching techniques, especially those leveraging libraries like OpenCV and custom stitching libraries, offer convenience and efficiency in stitching large sets of images with minimal manual intervention.
- Feature detection and matching algorithms, combined with camera estimation and adjustment techniques, play a crucial role in accurate image alignment and stitching.

- Image warping, blending, seam finding, and exposure error compensation are essential steps in the stitching pipeline, ensuring seamless integration of input images and minimizing visible artifacts in the final panorama.
- The choice of stitching technique depends on factors such as the complexity of the scene, the number of input images, computational resources, and the desired level of manual intervention.
- Finally we are able to stitch images using method1 and method2 and are able to implement these method to do video stitching as well.

References:

https://github.com/OpenStitching/stitching

https://arxiv.org/pdf/1802.04645v2.pdf

https://paperswithcode.com/paper/natural-image-stitching-with-the-global

https://www.researchgate.net/publication/267638940_Image_Stitching_based_on_Feature_Extraction_Techniques_A_Survey

https://openaccess.thecvf.com/content/CVPR2022/papers/Nie_Deep_Rectangling_for_Image_Stitching_A_Learning_Baseline_CVPR_2022_paper.pdf

Thank You