University of Nottingham
UK | CHINA | MALAYSIA

COMP3065: Computer Vision Coursework Report

Binhe LI (20320527)

May 6, 2024

words count: 2473

School of Computer Science
University of Nottingham Ningbo China

Contents

1 Introduction

The main goal of this project is to stitch the video frame by frame into a panoramic image by an appropriate method. In order to increase the functionality and completeness of this project, we added software interface design, image parameter adjustment, and picture black part cropping.

2 User Interface(UI)

2.1 Aim

This feature was added to facilitate the use of this program by the user. The user can select the file path of the video and the location of the image through the UI. This also makes it easy for the user to view the image, adjust its parameters, and see the image after cropping.

2.2 Implementation

We implemented the UI design and implementation based on Tkinter GUI library in Python. The whole function is divided into several key parts: selecting video files to generate panoramic images, editing panoramic image parameters, and cropping the black part of the image. Here's a breakdown of the logic and implementation:

2.2.1 Main modules and libraries

tkinter: Used to build graphical user interfaces threading: Use multithreading to avoid blocking the GUI when processing the video. filedialog, messagebox, ttk: For file selection dialog, message prompt, and improved Tkinter component styles, respectively.

2.2.2 Functional Implementation Details

Main window Settings In the main window, the user can select the video file, start the processing, and view the processing progress.

Select the video file The select_video function allows the user to select a video file through the file selection dialog box. The file path is stored in the global variable filename and displayed in the GUI.

Start video processing The start_processing function first checks whether the file has been selected and displays an error if it has not. threading. Thread is used to start the video processing, ensuring that the GUI does not freeze in the process. During processing, the process_video function is called to generate the panoramic image, and a new function is called to generate a new interface to display the image after the panoramic image is generated.

Move the canvas Since panoramic images are large, the move_canvas function allows the user to move the image up, down, left and right in the GUI to view different parts of the image.

picture black edges trimming By clicking the "Trim Black Edges" button, the user can immediately see the picture with the black edges removed. The user interface also supports user-defined parameters and directions for clipping.

Update the GUI and progress bar update_gui is used to update the GUI interface while processing a video, for example to display processed images or to update progress information. update_progress is used to update the progress bar.

Image editing function Create a new window in open_editor for users to edit the image, providing adjustments for brightness, contrast, blur, saturation, and sharpening. Use the slider to control the parameters of image processing, and preview the effect in real time after adjustment.

Save the image The save_image function allows you to save an edited image to a specified path.

2.3 Output

This is the main window of the user interface after launching the program

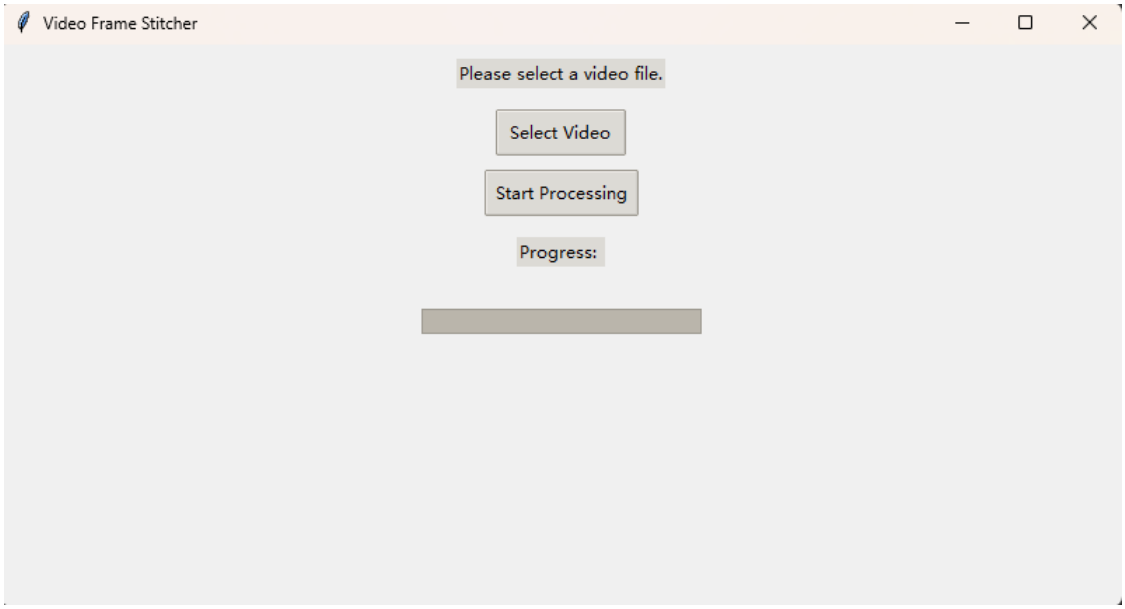

Figure 1: Main Window

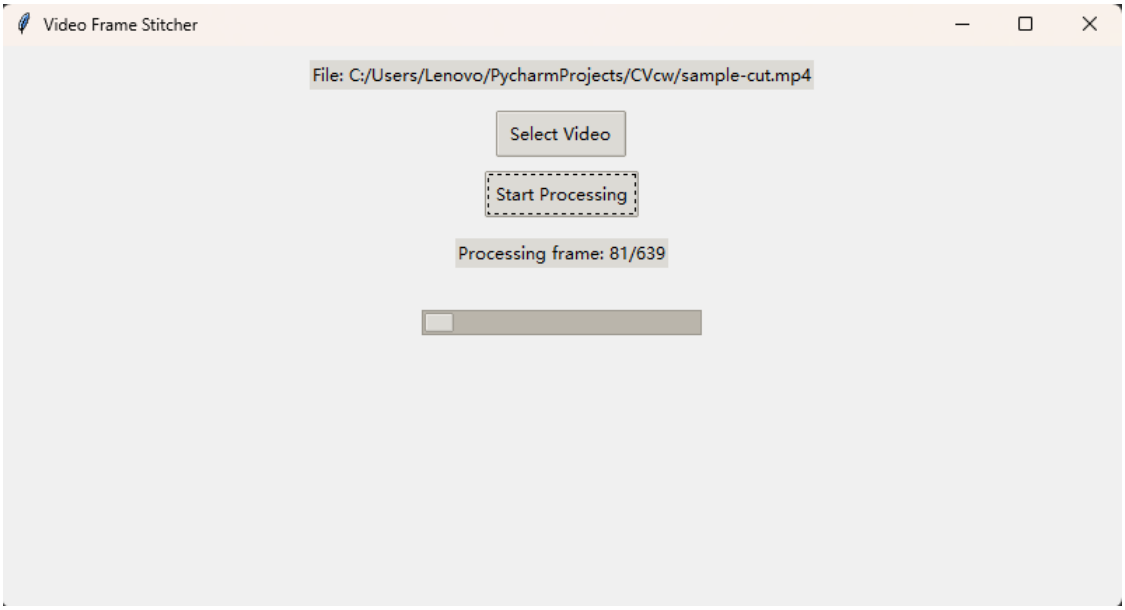

Figure 2: Interfaces of video processing

Shows the path of the processed video and the progress of the image generation

Figure 3: An interface to display images

The user can select the view to move the image, edit the image, or select a specific threshold and orientation to crop the black edges of the image.

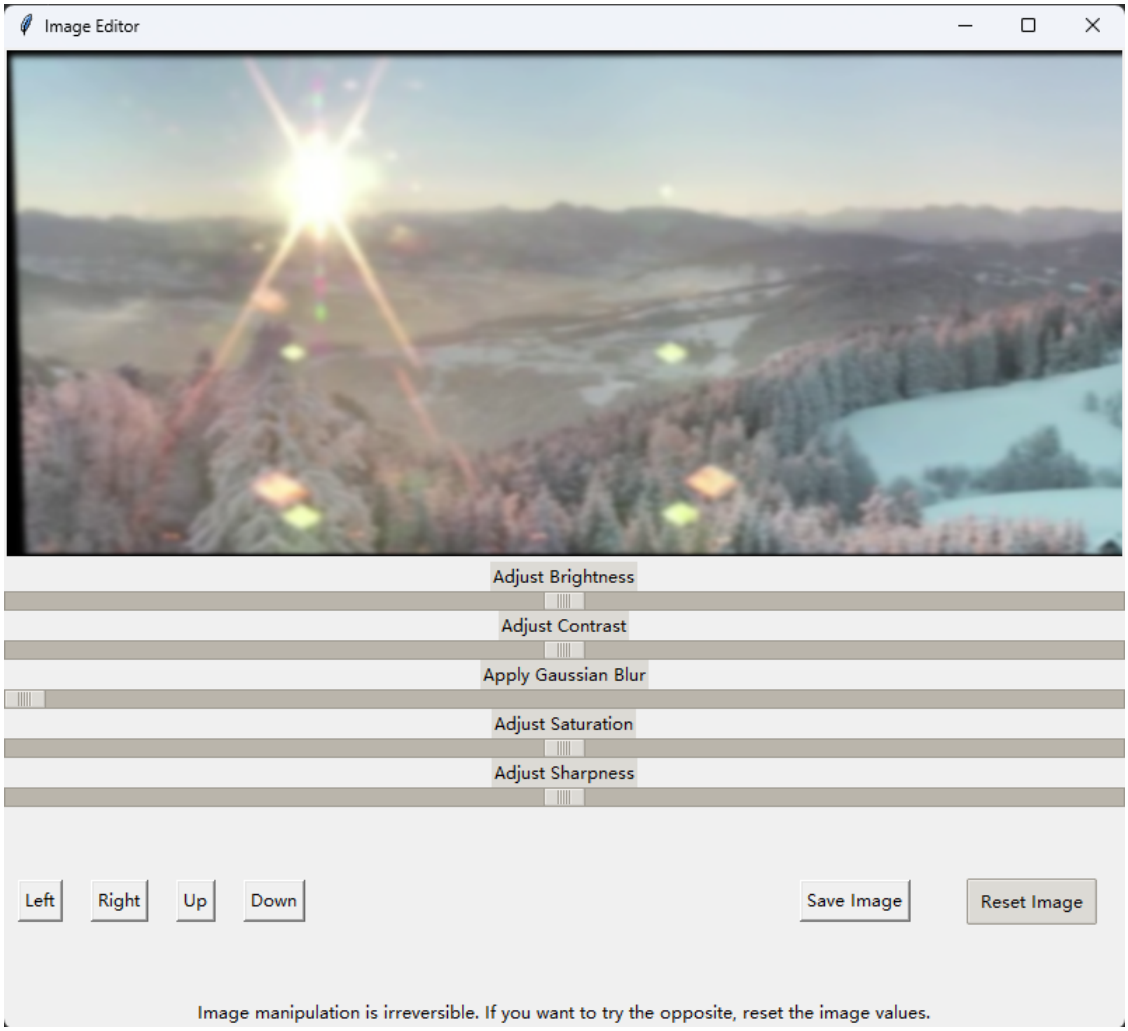

Figure 4: Image editor

After selecting the image editing screen, the user can change the image parameters, reset the image, or save the image.

2.4 Evaluation

2.4.1 Strengths

Simple and intuitive operation The user interface is simple. Selecting files, starting processing and saving images, etc. can be done through the obvious buttons in the interface. Explicit prompts and progress updates make the user aware of the current state of the application.

Multiple interface The main page is responsible for selecting the video path and processing the video. The generated images are displayed in a new screen. The image editor and the prediction of the elements in the image are also shown in other Windows. It makes the user interface have good interactivity and integrity.

Real-time feedback Real-time display of processing progress through progress bars and status messages. At the same time, the instant preview in the image editing function allows users to see the adjustment results directly.

Multi-threaded processing Avoid the interface freezing when processing large files by using background threads for video and image processing.

Extensibility and maintainability By calling different functions to perform different functions and produce different interfaces, makes the code easy to maintain and extend.

2.4.2 Weaknesses

Limited interface aesthetics Tkinter's controls are visually simple, which may have an impact on the user experience. The interface layout is relatively basic and may appear crowded or unintuitive for the organization of complex operations.

Performance Optimization For very large video files or high-resolution images, Tkinter may not be sufficient to provide smooth performance. Although multithreading can avoid interface freezing, it may require more careful handling of resource management and thread synchronization to optimize performance.

2.4.3 Comparison of the initial expectations of this feature with the test results

The purpose of the user interface is to make it easy for the user to run the function of the program. Because of the variety of the interface, the running of the program is sequential and logical. This effectively prevents users from incorrectly running features such as "open image editor before image is generated". Although the whole UI logic doesn't perfectly handle all user actions. But the simplicity and interactivity brought by the user interface makes it easy for users to run the functions implemented in this project.

3 Panoramic image generation

3.1 Aim

The purpose of this function is to convert a video file into a corresponding panoramic image. The program reads partial frames of the video at regular intervals. These frames are then concatenated using specific computer vision techniques to generate a wide-view panorama.

3.2 Implementation

3.2.1 Initialization and configuration

The following two functions are derived from homemade log files to catch exceptions:

Logging setup First, the setup_logging function is called to configure the logging for tracking application status and debugging information.

Video file check The check_file function is used to verify that the specified video file exists and is readable. About Libraries used We use the opencv library for the video or image processing and numpy for the data processing.

3.2.2 Reading video frames

Open the video file using the VideoCapture method of the opencv library. The total number of frames in the video is obtained by using the CAP_PROP_FRAME_COUNT of the opencv library. Set up a loop to read the video frame by frame. Each time through the loop, the current frame and the previous frame are synthesized into a single image.

3.2.3 Feature detection and matching

Feature detection SIFT (Scale-Invariant Feature Transform) algorithm is used to detect keypoints and descriptors. SIFT is able to provide features that are invariant to rotation and scale changes. Due to the change of shooting Angle or distance from the object in the video. SIFT makes it possible to compare keypoints and descriptors from two frames and splice the two images accordingly.

Feature matching The FLANN (Fast Library for Approximate Nearest Neighbors) matcher is used to match the descriptors of the two images. The matching process is optimized by setting the algorithm and search parameters (index_params and search_params). We apply Lowe's ratio test to filter the matches and keep only the high quality matches.

3.2.4 Image alignment and stitching

Compute the homography matrix Compute a homography matrix using the matched keypoints (opencv.findHomography). This matrix is used to transform the current frame to the view of the previous frame to achieve the same Angle stitching of two images.

Perspective transformation With the opencv.warpPerspective function, we apply a perspective transformation to the current frame using the homography matrix obtained in the previous step to align it with the previous frame.

3.2.5 Image fusion

Weighted fusion The weighted average method is used to fuse the images in the stitching area, which makes the visible gaps in the stitching less and

the transition more natural.

Detect and crop The final panoramic image is grayed and binarized to identify non-black areas in the image. The methods cv2.findContours and cv2.boundingRect are then used to find the smallest rectangle that contains all non-black pixels and crop the image according to this rectangle to remove the black edges caused by the stitching.

3.2.6 Result output and cleanup

Save and callback The processed panoramic image is saved to a file, and the image path and status information are returned through the callback function to make the UI update in time and wait for the next step.

Resource release After the video is processed, release the video files and other resources to keep the program clean and stable.

3.2.7 Performance optimization and error handling

Multi-frame interval reading To optimize processing time and resource consumption. The program chooses to process every 20 frames, which can reduce the computational burden.

Error and exception management In key steps such as file checking, feature matching, homography matrix computation, we have set up appropriate exception catching. When an error occurs, appropriate feedback is sent back to the console to debug or modify the flow.

3.3 Output

Figure 5: Panoramic picture

This image is a panoramic image generated from a 21 second landscape video [1]

3.4 Evaluation

3.4.1 Strengths

Complexity and Innovation of technical implementation The SIFT algorithm is used to detect and describe the local feature points. SIFT algorithm is a classical and powerful feature extraction method in computer vision. It shows good invariance to rotation and scale changes of photos, so it is suitable for the generation of panoramic images. FLANN matcher is used to find the nearest neighbor point quickly and efficiently to improve the matching efficiency and speed.

Image alignment and fusion technology Homography matrix is used for image alignment to realize the Mosaic of images from different views. The weighted fusion technique at image stitching reduces the visibility of stitching lines and improves visual continuity.

Process feedback in real time Real-time progress feedback ensures that the user can track the processing progress. When working with large files or long videos, the appearance of progress can especially enhance the user experience in the application.

3.4.2 Weaknesses

Processing time and resource consumption Although FLANN improves the matching efficiency, SIFT feature extraction is still computationally intensive. This adds significant resource consumption and processing latency to the program. Although the computation of the homography matrix is necessary in image alignment. But when the number of feature points is huge, the matrix calculation is also the main part of the resource consumption.

Matching accuracy issues Feature matching is easy to make mistakes in scenes with few feature points or many repeated textures, which leads to

errors in image stitching. In scenes with large illumination changes, the SIFT algorithm may not guarantee a stable matching effect.

3.4.3 Comparison of the initial expectations of this feature with the test results

The expectation of the design is to generate continuous and visually coherent panoramic images through the extraction of video frames and the use of specific computer vision techniques. In most cases, the program succeeds in this goal. However, when dealing with long or high-resolution videos, the problem of long program processing time and large resource consumption is still significant. Moreover, the more a certain frame of the video is processed first, the more blurred the corresponding part will be at the end. On the far right, the stitching is also not complete due to the number of feature points or descriptor matching problems.

4 Image parameter adjustment

4.1 Aim

The purpose of this function is to provide users with a graphical user interface to adjust the contrast, brightness and other parameters of the panoramic picture.

4.2 Implementation

4.2.1 Graphical user interface construction

Use the tkinter library to create the main window and an edit window. In the editing window, multiple sliders are configured for image adjustment operations to adjust values such as brightness, contrast, saturation, sharpness, and blur. Use the Python Imaging Library (PIL) to load the images. The image is displayed on tkinter's Canvas so users can see the editing in real time.

4.2.2 Implementation of image editing function

For the adjustment of brightness, contrast, sharpness, and saturation, the ImageEnhance module of PIL was used. The Gaussian blur is implemented

using ImageFilter.GaussianBlur. All editing operations are triggered by adjusting the value of the slider, and the image preview is instantly updated.

4.2.3 Moving and saving images

Use the tkinter library to provide directional buttons (left, right, up, down) to move the image viewport. The Save button allows the user to save the edited image locally.

4.2.4 Image reset function

Providing a reset button allows the user to reset the image to the state it was originally loaded in by keeping the initial image variable in advance.

4.3 Output

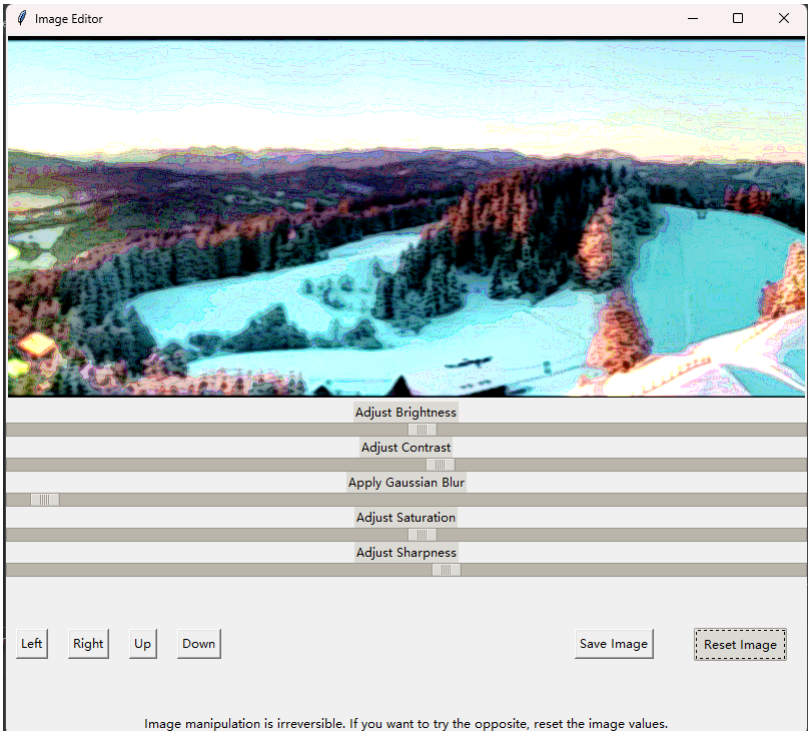

Figure 6: Parameter tuning

By adjusting these parameters, we can make adjustments to the image.

Figure 7: Final result

Different parameter adjustments can make the image have a different style or improve the sharpness.

4.4 Evaluation

4.4.1 Strengths

User interaction Sliders and buttons make it easy for users to adjust image attributes.

Live preview Any editing operation is instantly visible on the image preview.

Rich in features The editing of a variety of image parameters meets the basic needs of image processing.

4.4.2 Weaknesses

Functional Limitations Lack of more advanced image editing features such as layer manipulation, color correction, etc. The image movement function is relatively basic.

User interface aesthetics The interface, while functional, can seem too rudimentary in a modern application.

4.4.3 Comparison of the initial expectations of this feature with the test results

The initial design goal was to create a tool that would provide basic image editing functionality. From the practical point of view, the program basically

achieves the expected goal. However, from the point of view of extensibility and aesthetics, there is room for improvement.

5 Black cropping

5.1 Aim

Due to the jitter of the perspective of the recorded video, the edge of the panoramic image generated by stitching has a black edge caused by stitching. The purpose of this program is to automatically detect and crop black edges in an image to improve the visual effect and the efficiency of subsequent image processing.

5.2 Implementation

5.2.1 Image loading

We use the image.open method of PIL to load the Image file and the numpy library to convert the image data into an array format for numerical calculations.

5.2.2 Black pixel detection

The proportion of black pixels in each column and row of the image is calculated. Here, a black pixel is defined as a pixel value less than or equal to a specified threshold (black_threshold), which defaults to 5. If the proportion of black pixels in the current row or column is greater than the given ratio (black_ratio_threshold, which defaults to one-third), then the row will be cropped.

When the black area is too large, it may cause most of the effective area of the picture to be cropped incorrectly, so a specific cropping can be performed by inputting the direction selected by the user.

5.2.3 Result processing

The cropped image is stored in the default path "images/result_cropped.png", after which the user interface reloads the image and displays it.

5.3 Output

Figure 8: Image before cropping

This image is from a video taken in my bedroom at a 90 degree Angle. Since my shot was not steady, the video content was not on a horizontal line, resulting in black edges.

Figure 9: Image after cropping

After cropping, the panoramic image becomes more clean and tidy.

Figure 10: Difficult situation

In this case with a large black area, the right region of the picture is dominated by black in the vertical direction. This causes the valid part of the right side to be cropped as well.

Figure 11: Image after cropping

In this case, we can select a specific direction to filter the valid pixels, such as selecting only the horizontal direction.

5.4 Evaluation

5.4.1 Strengths

Automating The fully automated process reduces the need for manual editing and results in cleaner panoramics.

High flexibility The threshold and scale can be adjusted according to the actual needs to be suitable for different image processing scenarios.

Accuracy By accurately calculating the proportion of black pixels and the threshold, we can effectively determine the boundaries that need to be cropped.

5.4.2 Weaknesses

Flexibility Limitations The default parameters are not suitable for all images. The user needs to adjust the parameters for different images.

5.4.3 Comparison of the initial expectations of this feature with the test results

The initial design goal was to create a tool that would automatically crop the black edges of an image to improve the visual effect of the image. From

the results, the program basically achieved the expected goal. However, the adjustment of the relevant threshold parameters limits the ease of use of the procedure.

6 References

1. https://www.youtube.com/watch?v=VDTEyQhZzKA&t=0s