Jose Hernandez

Professor Vaca

CAP 4453

26 FEB 2025

Project 1: Documentation

1. Manual Mode:

Manual mode document scanning requires the user to input the coordinates of the origin point of the image to warp manually. This introduces additional requirements, as the points must be entered in logical order: A-B-C-D, where A is the starting point and B is the next corner in a counter-clockwise direction. Please follow the provided coordinates and input them in the order given. Failing to do so will result in incorrect image warping.

To create manual mode, I used my compute implementation from Homework 5, but I leveraged OpenCV to perform the warping. My method from Homework 5 did not produce a perfectly smooth result, possibly because I used nearest neighbor interpolation to estimate the missing pixels.

After selecting the four corner points, the program will provide visual confirmation of the selected points.

Finally, I also implemented a function to rotate and flip the image to get the desired orientation, as the sample images may be correctly warped but might have an orientation that makes it difficult to appreciate their content.

'Img0'

Points A: (x110, y995)
Points B: (x475, y1237)
Points C: (x1000, y420)
Points D: (x655, y205)

- Rotate 90 degrees: option 1
- Flip vertically: option 5
- 0 to proceed

Instructions:

1. Enter coordinate X for point A: input 110, press enter
2. Enter coordinate Y for point A: input 995, press enter
3. Enter coordinate X for point B: input 475, press enter

4. Enter coordinate Y for point B: input 1237, press enter
5. Enter coordinate X for point C: input 1000, press enter
6. Enter coordinate Y for point C: input 420, press enter
7. Enter coordinate X for point D: input 655, press enter
8. Enter coordinate Y for point D: input 205, press enter
9. Rotate 90 degrees: input 1, press Enter
10. Flip vertically: input 5, press Enter
11. Exit: input 0, press Enter

'Img1'

Points A: (x55, y1130)
Points B: (x1032, y1795)
Points C: (x1502, y1092)
Points D: (x512, y413)

- Rotate 90 degrees: input 1, press Enter
- Flip horizontally: input 4, press Enter

1. Enter coordinate X for point A: input 55, press enter
2. Enter coordinate Y for point A: input 1130, press enter
3. Enter coordinate X for point B: input 1032, press enter
4. Enter coordinate Y for point B: input 1795, press enter
5. Enter coordinate X for point C: input 1502, press enter
6. Enter coordinate Y for point C: input 1092, press enter
7. Enter coordinate X for point D: input 512, press enter
8. Enter coordinate Y for point D: input 413, press enter
9. Rotate 180 degrees: input 2, press Enter
10. Flip horizontally: input 4, press Enter
11. Exit: input 0, press Enter

Conclusion:

Manual mode is an effective way to warp an image by visually identifying its corners. This approach significantly reduces computational time compared to automated corner detection, which can be complex due to the extensive preprocessing required. For example, automated detection would necessitate applying filters to enhance edge detection, techniques to refine edge thickness, and background removal. However, relying solely on manual input is impractical for real-world applications, as it introduces additional steps and requires user intervention.

2. Fully Automated Document Recovery:

The link to the image provided by the professor is:
https://raw.githubusercontent.com/joseghm1/hw_images1/refs/heads/main/receipt.jpg

Please use this link to input the image for testing this part of the assignment. The image is hosted in my github account. However, if you have the image hosted anywhere else and it exits in .jpg format it can be used. I just hosted the image since it would generate an error if I try to get the image directly from the reddit forum.

Instructions:

1. Run the program
2. Input this link: https://raw.githubusercontent.com/joseghm1/hw_images1/refs/heads/main/receipt.jpg
3. Hit enter.

For this part of the program, I implemented the following steps:

1. Converted the image to grayscale.
2. Blurred the image using a median filter with a sigma of 23.
3. Used the Canny edge detector with threshold values of 70 and 110.
4. Found contours using cv2.findContours and sorted them from largest to smallest.
5. Approximated the largest contour into a polygon. If the contour had more than four vertices, I used the gift wrap algorithm (Convex Hull).
6. Applied a perspective transform to align the document correctly.
7. Warped the image to produce the scanned document.

Explain the idea behind your method:

The overall idea of this method was to identify the document to scan by obtaining the largest contour. To achieve this, I blurred the image to reduce noise from the content inside the receipt, as it could interfere with detecting the document's overall shape. Then, I used the Canny edge detector to capture the document's boundaries. After obtaining various contours, the largest one, which corresponds to the document, was selected. The corners were then determined by finding the extreme points of this contour. Finally, I performed a perspective transform using these points to obtain a properly warped image.

Do you have other ideas in How could make your algorithm better?

To improve this algorithm, I would have used Gaussian filters to sharpen the edges of the document. This would have created a better contrast with the background. Additionally, I didn't use binarization techniques like Otsu's method, but I believe that if the right filter had been applied, it could have improved the detection of the document's contour by converting the image to black and white.

Another enhancement would be to use morphological operations, like dilation, to close small gaps in the edges, which could appear at the threshold level. These gaps might help eliminate the

noise for some letters but negatively affect the overall edge, making it harder to detect the four corners of the document.

Conclusion:

The implementation of automated document recovery is effective for scanning documents with sharp edges and high-contrast backgrounds, such as the sample image provided for this assignment. However, its performance may decline when dealing with documents with thin edges or wrinkled surfaces. Several techniques could enhance the algorithm's robustness, including automatic threshold adjustment, binarization methods, morphological operations, and improving the initial perspective before corner detection. While further refinements are needed, this implementation serves as a strong foundation for developing a document scanning tool, requiring only minor adjustments for broader applicability.