Image Stitching and Disparity Map Estimation CS 554 Homework

Melih Berk Yılmaz, *Member, Bilkent University,* Fuat Arslan, *Member, Bilkent University,*

I. Introduction

THIS report presents the implementation of image stitching and disparity map estimation using classical computer vision techniques. The first part of the report covers the methodology and results for image stitching, while the second part focuses on the methodology and results for the disparity map estimation.

To stitch the given images, we initiated the process by extracting the features of each image using SIFT. These features were then matched to find their corresponding counterparts. However, due to the potential errors in the SIFT algorithm, we applied the RANSAC algorithm to eliminate outliers, resulting in a set of highly robust matched features. Utilizing these features, we estimated the homography matrix and performed image warping from one to another. Finally, we combined these two images using various techniques, such as alpha blending and seam finding.

In the second stage of this report, disparity map estimation was employed. In this section, a rather simplistic approach, as depicted in the homework, was utilized. Assuming rectified images, we selected a patch from the left (or right) image. Subsequently, another patch was slid along the corresponding rows of the right (or left) image. By assessing the normalized cross-correlation between these patches, we determined the best match and calculated the disparity.

II. Image Stitching

A. Feature Extraction and Matching

Firstly, the SIFT algorithm is utilized to extract feature points from each image, generating 128D descriptor vectors for each feature points. To match corresponding image feature points between two images, the Nearest Neighbor Distance Ratio (NNDR) approach is employed. This method involves listing the distances of features in the first image to descriptors in the second image and computing the ratio between the closest distance and the second closest distance. Matches are determined based on a threshold, typically set at 0.7 in our scenario.

The NNDR approach proves more effective compared to the 1-Nearest Neighbor method as it avoids matching all features when distances are excessively high. Additionally, it outperforms the Euclidean distance approach by not necessitating a threshold for selecting the best matches, thus preventing the selection of multiple matches. While NNDR does have its own threshold, it serves more as a regularization term and doesn't lead to the aforementioned issues.

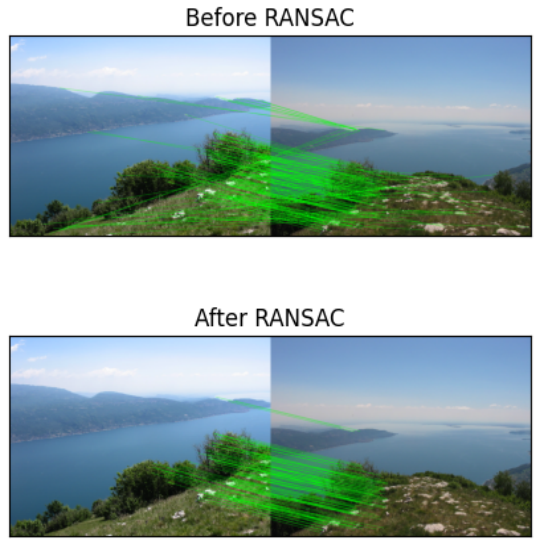


Fig. 1: Feature Matching Output

B. RANSAC and Homography Finding

The SIFT algorithm, as mentioned earlier, is prone to errors. To mitigate these issues, we've implemented the RANdom SAmple Consensus (RANSAC) algorithm, which helps us obtain more robust and reliable feature matches. As seen 1 that, algorithm managed to get rid of some bad matches.

The images we're stitching together have different viewpoints, and before stitching them, we must ensure that their intersection points are aligned. To achieve this, we need to create a mapping from one image to another, which is where homography comes into play. Homography is a mathematical transformation that describes the relationship between two different 2D projective geometric spaces. It's represented by a 3x3 matrix and is used to map points from one plane to another. Thanks to RANSAC, we've obtained robustly matched image points. While making outlier elimination as aforementioned, it also finds best homography by using random sampling.

In the case of homography, there are 9 unknown parameters, but due to the use of homogeneous coordinates, there are effectively 8 parameters to determine. This means that theoretically, we only need 4 equations or, in other words, 4 point matches to determine these parameters. Using these points, we can calculate the homography using a least squares solution. Here's the formulation for this calculation:

$$\begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1u_1 & y_1u_1 & u_1 \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1v_1 & y_1v_1 & v_1 \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2u_2 & y_2u_2 & u_2 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2v_2 & y_2v_2 & v_2 \\ -x_3 & -y_3 & -1 & 0 & 0 & 0 & x_3u_3 & y_3u_3 & u_3 \\ 0 & 0 & 0 & -x_3 & -y_3 & -1 & x_3v_3 & y_3v_3 & v_3 \\ -x_4 & -y_4 & -1 & 0 & 0 & 0 & x_4u_4 & y_4u_4 & u_4 \\ 0 & 0 & 0 & -x_4 & -y_4 & -1 & x_4v_4 & y_4v_4 & v_4 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = 0 \quad (1)$$

Solving equation (1) using a least squares solution provides us with a candidate homography. However, we enhance the reliability of this homography by employing RANSAC. RANSAC involves random sampling of 4 points to find the best homography, resulting in the maximum number of inlier matches. To limit the number of iterations, we've set a boundary condition of 100 inliers as a stopping point, and the maximum iteration number is empirically set to 100,000. This approach helps us obtain a robust and accurate homography for aligning the images before stitching them together.

Fig. 2: Sample Warping Algorithm Output

To attain image alignment using acquired homography one image warped to other one. A sample of warping output can be seen from 2 that second image warped to first image plane such that their intersection aligned.

C. Image Blending

Up to the point of blending, we have successfully determined the best homography using the RANSAC algorithm, which allows us to align the images. However, simply overlaying one image on top of the other at their intersection points may result in unnatural-looking images. This unnatural appearance can be attributed to variations in illumination, contrast differences, and other effects. A sample of this effect can be seen from Figure 3.

To address this issue, we introduce the concept of blending. Blending serves as a solution to the problem of unnatural-looking intersections. Instead of previous approach blending compose an intersection using both images. One common approach is alpha blending.

Alpha blending introduces a weighting mechanism between two images. Let's define $\alpha(x, y)$ such that its linear in x. It starts at 1 on the intersection boundary closest to the first image and becomes zero at the edge closest to the second image. $1 - \alpha(x, y)$ is the weight of the second image. The equation below illustrates how the brightness of the intersection is mathematically determined:

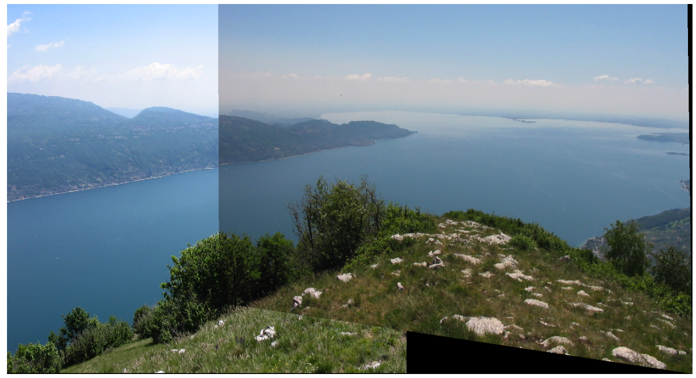


Fig. 3: Sample Output without Blending

$$I_{\text{res}}(x, y) = \alpha(x, y)I_1(x, y) + (1 - \alpha(x, y))I_2(x, y) \qquad (2)$$

The weighting function $\alpha(x, y)$ maintains its value beyond the intersection boundaries.

D. Results for Image Stitching

For the image stitching process, specific parameters were employed: an NNDR (Nearest Neighbor Distance Ratio) threshold of 0.7, 100,000 iterations for RANSAC, and a maximum of 100 inliers. Visually, the results obtained were satisfying.

However, there is a discernible issue with the clarity of the warped images, stemming from the interpolation method used in the custom warping function. The current implementation utilizes linear interpolation, resulting in a slight blurriness within the filled pixels. The blended output can be observed in Figures 4, 5, 6.

III. Disparity Map Estimation

A. Normalized Correlation Matching

The algorithm for estimating the disparity map relies on matching pixels one-to-one, assigning each pixel a specific disparity value. We operate under the assumption that the test images have been rectified, meaning that their epipolar lines are horizontal. Additionally, we assume that the maximum disparity between image pairs does not exceed 65 pixels.

Pixel-to-pixel matching presents challenges because a pixel value might appear multiple times in different parts of the other image. To address this, we chose a specific window size and computed the correlation between these windows. Initially, both the left and right images are normalized, and correlations between patches are calculated to find the best match. Subsequently, we experimented with normalizing only the selected patches and observed that this approach yielded superior results.

Consequently, we proceed by selecting a batch from the left (or right) image denoted as w_L, then iterate over the corresponding rows of the right (or left) image denoted as w_R. These patches are normalized using Equation (2):

$$\bar{w} = \frac{w - \mu_w}{\|w - \mu_w\|_2} \qquad (2)$$

Fig. 4: Blended Output of The First Example

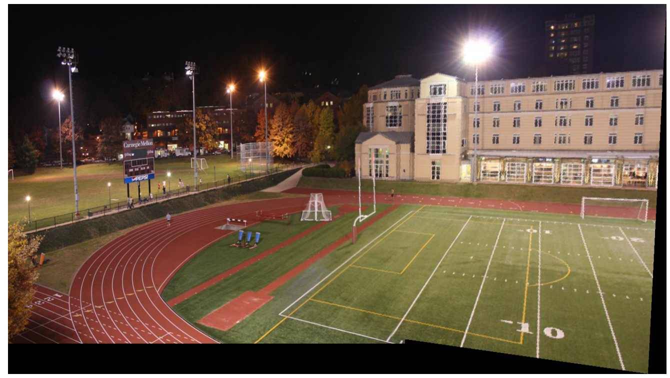


Fig. 5: Blended Output of The Third Example

Fig. 6: Blended Output of The Second Example

The correlation between the two patches is calculated using Equation (3), aiming to find the pixel that maximizes correlation $C(d)$:

$$C(d) = \bar{w}_L(x, y) \cdot \bar{w}_R(x - d, y) \quad (3)$$

where d^* is determined by:

$$d^* = \arg\max_d C(d) \quad (4)$$

B. Results for Disparity Map Estimation

The sole parameter influencing the estimation of the disparity map is the window size, which dictates the level of noise and detail in the output image. As the window size increases, noise diminishes, yet this also leads to a loss of finer details, resulting in increasingly smoother outcomes.

Through experimentation with various window sizes, we discerned the most optimal ones based on visual appearance. This determination was made due to the substantial difference in the range of disparity maps generated, with a maximum value of 65, in contrast to the ground truth maps that extend up to 200. The optimal window sizes determined for clothing photos were found to be 17x17, while for plastic photos, the most effective dimensions were 19x19.

As demonstrated in the figure depicting the disparity maps for plastic and cloth images (Figure 7, 8), the predicted and ground truth disparities for both left and right frames are visually compared.

IV. Implementation Details

In this report, we have developed and implemented image stitching and disparity map estimation using classical computer vision methods. Our implementation incorporates a combination of default functions from the cv2 library and custom-built algorithms.

The initial phase involves our implementation of the Feature Matcher and RANSAC algorithms. Subsequently, we devised a warping function utilizing linear interpolation and followed it with the implementation of the alpha blending algorithm. The second part of the report is entirely our own implementation, devoid of any default library functions. Here, we executed the complete process independently.

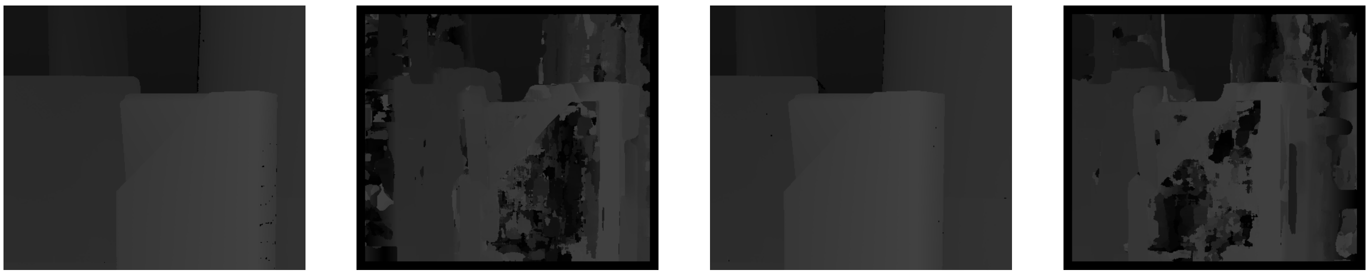
V. Conclusion

In conclusion, the implementation of image stitching and disparity map estimation using classical computer vision techniques has been successfully demonstrated in this report. The methodology employed for image stitching involved feature extraction using SIFT, followed by feature matching and outlier elimination using RANSAC, resulting in a robust set of matched features. The subsequent homography matrix estimation and image warping facilitated the seamless combination of images through techniques like alpha blending and seam finding.

For the disparity map estimation, a simplistic yet effective approach was utilized, assuming rectified images and employing normalized cross-correlation to determine the best match

(a) Ground Truth Disparity (Left) (b) Predicted Disparity (Left) (c) Ground Truth Disparity (Right) (d) Predicted Disparity (Right)

Fig. 7: Disparity Maps for Cloth Images



(a) Ground Truth Disparity (Left) (b) Predicted Disparity (Left) (c) Ground Truth Disparity (Right) (d) Predicted Disparity (Right)

Fig. 8: Disparity Maps for Plastic Images

and calculate the disparity. Overall, this report showcases the application of fundamental computer vision techniques in achieving image stitching and depth estimation, providing insights into the process and outcomes of these methodologies.