Yilong Wang, yw14218, 01362420 — Chenyang Wang, cw3918, 01497270

1. Collect Data

Task 1 The critical point of this task is to establish a proper dataset that could well handle subsequent computer vision tasks. To achieve this, the images in the dataset must consist mainly of a textured object with rich features and seldom background. Another challenge in this task is to avoid common photographic mistakes such as blurriness, overexposure and reflection. We used a teddy bear toy as the target of interest, with an overhead view and controlled lighting to ensure soft and clear images. We printed out a 7*6 chessboard grid [1] that has easily detectable feature points as a reference to the world coordinates. The full dataset with image resolution of 1279x1706 pixels can be found in Appendix 6.1 - 6.3.

2. Keypoint correspondences between images

Task 2 In this task, we compare quality and quantity of correspondences found by manual labelling using Paint 3D [2] and automatic labelling using brute-force matching with SIFT descriptors [3], which takes the descriptor of one feature in the first set and matches it with all other features in the second set via distance metrics. The challenge is conducting fair experiments and determining the correct metrics for comparison. We conducted two comparison experiments on the pair of images HG_4, HG_5 and the pair of images HG_3, HG_8 respectively.

1. **Comparison between HG_4 and HG_5:** This pair of images was taken from similar angles and possessed a high degree of similarity. Figure 1 and Figure 2 shows manually plotted and automatically generated results respectively. Both methods produce excellent results. However, the manual comparison could only identify prominent and semantically high-level feature points (e.g. English letters, creases and stitch corners). Thus, the number of feature points matched is limited, and pixel-level random errors can frequently occur. In this case, we could visually detect only around 40 matches with high confidence. On the other hand, automatically comparison is capable of detecting low-level feature points and enables comparison between feature points with tiny textural differences. In this case, the SIFT algorithm could match over 500 feature points in seconds with hardly any outliners observed, which suggests that the automatic comparison outperforms the manual comparison at both quantity and quality level in this case.

2. **Comparison between HG_3 and HG_8:** These two images were far less similar than the above case. Figure 33 and Figure 35 in Appendix shows manual comparison and automatically generated results respectively. Around 30 matches could still be visually detected with high confidence and matched and generally produce high-quality manual comparison results. However, the change in camera angle leads to severe contamination of the descriptors of low-level feature points, which results in poor detection results for automatic comparison. Even among the top 50 feature point matches, about half of them are outliers with significant systematic errors.

In summary, automatic detection outperforms manual detection in terms of quality, efficiency and quantity when high image similarity. When the similarity of the images becomes low, especially when there is a projective transformation between image pairs, the quality of automatic detection drops and becomes less effective than manual detection.

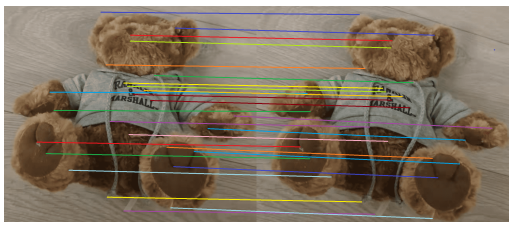

Figure 1: Manual Comparison between HG_4 and HG_5.

3. Camera calibration

Task 3.1 In this task, we found the camera parameters by using a 7*6 chessboard [1] via solving the homogenous linear system from correspondences using direct linear transformation algorithm[4]. The exact code used is OpenCV's C implementation of the camera calibration toolbox for Matlab [5]. The camera matrix intrinsic K is shown below. The focal lengths fx and fy are 1.264e+03, 1.258e+03. Since in a true pinhole camera model, the focal lengths in

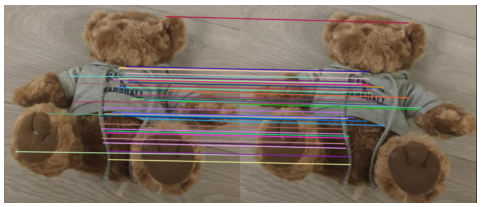

Figure 2: Automatic Comparison between HG_4 and HG_5. (top 50 matches plotted)

the two dimensions should be the same, this suggests there is a small error in calibration. The principle points x0, y0, which indicates the location of the principal point relative to the film's origin, are 6.332e+02, 9.054e+02. In this case, the vertical offset is higher than the horizontal offset. The skew coefficient is zero as the image axes are perpendicular [6]. The extrinsic parameters for the dataset FD with the grid are reported in appendix 6.2. For example, for the image FD_with_grid_1, the translation matrix is [[55.651], [204.468], [452.53]] and the rotation matrix is [[0.673], [-0.0247], [-3.01]]. These numerical results align with the fact that the relative position of the camera from the toy is rather high in depth, vertically down, and clockwise yawed.

$$K = \begin{bmatrix} 1.264e+03 & 0.00e+00 & 6.332e+02 \\ 0.000e+00 & 1.258e+03 & 9.054e+02 \\ 0.00e+00 & 0.000e+00 & 1.000e+00 \end{bmatrix}$$

Task 3.2 The distortion matrix (k1 k2 p1 p2 k3) is computed using cv.calibrateCamera the same as above [5]: [0.283, -1.310, 0.012, 0.008, 2.151]. Since the tangential distortion coefficients [7] p1 (0.012) and p2 (0.008) are neglectable, we only discuss the effect of radial distortion in this task. In order to illustrate camera distortions, we used OpenCV undistort function to generate an undistorted image of FD_with_grid_3 based on generated camera distortion coefficients. Figure 3 shows the result. We magnified the chessboard region for both the original and undistorted images, then connected their corresponding edge and corner points to draw desired rectangular grid maps. It can be noticed from the green circulated area that the original image has positive distortion as grids slightly bulge outwards. The distortion effect is mitigated after the original image is undistorted, but the error still exists.

4. Transformation estimation

Task 4.1 Homography matrix We chose the HG_1 and HG_3 image pair to implement this task. Keypoints location and their descriptors are automatically matched and then used to compute the homography matrix via the

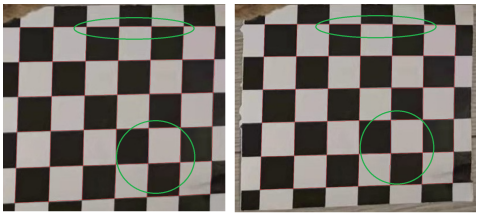

Figure 3: Comparison between distorted and undistorted chessboard of FD_with_grid_3. Left: undistorted. Right: original. Red shows their desired completely undistorted position of the grid. Green indicates places of interest.

internal OpenCV's findHomography function [8].

$$H = \begin{bmatrix} 1.115e+00 & 1.269e-01 & -1.381e+02 \\ 7.981e-02 & 1.138e+00 & -7.372e+01 \\ 8.920e-05 & 4.511e-05 & 1.000e+00 \end{bmatrix}$$

Since the photographed scene is not planar, we do not expect the homography matrix to measure the correspondence between these two images accurately. Figure 4 shows the correspondence results of manually annotated keypoints from HG_1 using approximation methods of projective transformation by affine transformation (ref lecture notes 2 page 15). Systematic errors can be clearly observed in the comparison results of those keypoints.

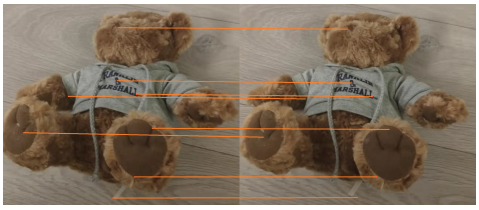

Figure 4: Computed keypoint correspondence results between HG_1 and HG_3 using estimated homography matrix

Task 4.2 Fundamental matrix Given that all images that belong to the same dataset are stereo, it is possible to compute a fundamental matrix and draw epipolar lines to compute more accurate keypoint matching correspondence. We chose FD_without_grid_3 and FD_without_grid_6 image pair to perform this task; the fundamental matrix is computed by the internal OPENCV findFundamental function.

$$F = \begin{bmatrix} 1.775e-06 & 8.298e-06 & -5.685e-03 \\ -3.508e-06 & -8.885e-07 & 2.199e-02 \\ -2.445e-03 & -2.127e-02 & 1.000e+00 \end{bmatrix}$$

Figure 5 shows keypoint correspondence results and their epipolar lines. We can notice that most feature points lie on

the epipolar lines that are computed using the estimated fundamental matrix, which indeed indicates a significant drop of correspondence systematic error compared to estimation using homography matrix. Please check Figure 36 in Appendix for the position of epipoles.

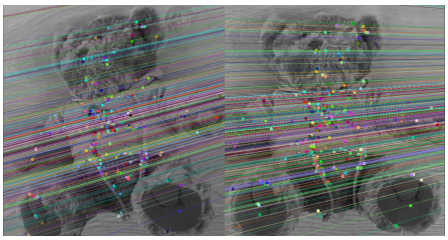

Figure 5: Computed keypoint correspondence and epipolar line results between FD_without_grid_3 and FD_without_grid_5 using estimated fundamental matrix.

We chose an image taken from another view to plot the position of vanishing points and their horizon line, and Figure 6 shows the plotted result.

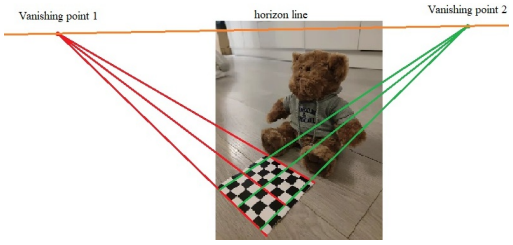

Figure 6: Vanishing points and the horizon line plot.

Task 4.3 Evaluation of Outliers In this task, we computed the fundamental matrices between different image pairs in the FD_without_grid data group and evaluated their matching performance. An estimation matrix is correct only if the keypoints of the bear's whole body were accurately compared and the direction of epipolar lines generally makes sense. From the experimental results in table 1, our estimation may tolerate approximately 50% of outliers.

5. 3D geometry

Task 5.1 Stereo rectified image pair We used two methods to generate stereo rectified pair of images. The first method utilises the OPENCV internal stereoRectifyUncalibrated function to generate rectified image pairs directly from generated keypoint matches and fundamental matrix in task 4.2. The second method uses the camera intrinsic matrix computed in task 2.1. Using the OpenCV internal

Test_index	Outlier_ratio	Estimation_correctness
Test_1	0.162	Correct
Test_2	0.31	Correct
Test_3	0.432	Correct
Test_4	0.494	Correct
Test_5	0.618	Incorrect
Test_6	0.732	Incorrect

Table 1: Relationship between outlier ratio and correctness of fundamental matrix estimation.

stereoCalibrate and stereoRectify [9], the translation vector, rotation matrix, essential matrix, and fundamental matrix could be computed and then used to calculate projection matrix and rotation matrix for each camera. According to the epipolar lines shown in figure 7 below and figure 37 in Appendix, both methods produce generally accurate results. "StereoRectifyUncalibrated" does not require the camera's intrinsic matrix to perform camera rectify but highly relies on the quality of keypoint matching results, while "StereoRectifycalibrated" performs stereo rectify based on the calibration grid only and could produce generally accurate result for any stereo image pair with lower similarity.

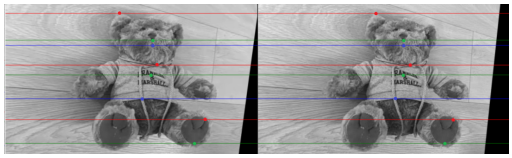

Figure 7: Stereo rectify result between FD_without_grid_7 and FD_without_grid_8

Task 5.2 Depth map Figure 8 shows the computed depth map of image FD_without_grid_7 generated from Figure 37 image pair using OPENCV StereoBM_create module, with brighter area indicating near pixels and darker area indicating pixels that are far away. This depth map recognises the head, hand and leg area closer to the camera. However, it can not distinguish between body area and background due to the lack of detectable background keypoints. Random errors could be observed but generally have small impact.

Figure 8: Depth map of FD_without_grid_7.

References

[1] Camera calibration grid opencv, 2022. https://docs.opencv.org/3.4/dc/dbb/tutorial_py_calibration.html.

[2] Paint 3d, 2022. https://www.microsoft.com/en-gb/p/paint-3d/9nblggh5fv99activetab=pivot:overviewtab.

[3] Brute-force matcher with sift descriptor, 2022. https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html.

[4] Krystian Mikolajczyk. Camera calibration, computer vision and pattern recognition, 2022. Lecture 3, slides 36/43.

[5] Jean-Yves Bouguet. Camera calibration toolbox for matlab, 2022. http://www.vision.caltech.edu/bouguetj/calib_doc/.

[6] J. Heikkila and O. Silven. "a four-step camera calibration procedure with implicit image correction", 1997. IEEE International Conference on Computer Vision and Pattern Recognition.1997.

[7] camera-calibration, 2022. https://www.mathworks.com/help/vision/ug/camera-calibration.html.

[8] feature_homography, 2022. https://docs.opencv.org/3.4/d1/de0/tutorial_py_feature_homography.html.

[9] opencv. "group__calib3d", 2022. https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html.

6. Appendix

6.1. Dataset FD without grid:

Figure 9: FD_without_grid_1

Figure 10: FD_without_grid_2

Figure 11: FD_without_grid_3

Figure 12: FD_without_grid_4

Figure 13: FD_without_grid_5

Figure 15: FD_without_grid_7

Figure 14: FD_without_grid_6

Figure 16: FD_without_grid_8

6.2. Dataset FD with grid:

Figure 17: FD_with_grid_1
Translation matrix : [[55.651], [204.468], [452.534]]
Rotation matrix : [[0.673], [-0.0247], [-3.013]]

Figure 18: FD_with_grid_2
Translation matrix : [[90.001], [204.527], [486.496]]
Rotation matrix : [[0.434], [0.0367], [-3.0865]]

Figure 19: FD_with_grid_3
Translation matrix : [[93.145], [214.885], [472.035]]
Rotation matrix : [[0.240], [-0.0156], [-3.103]]

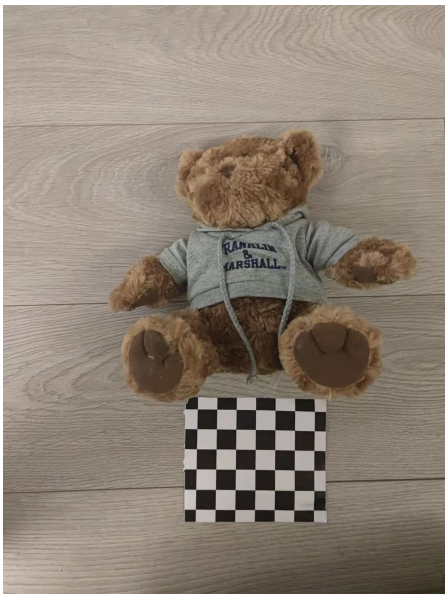

Figure 20: FD_with_grid_4
Translation matrix : [[92.858], [196.312], [460.246]]
Rotation matrix : [[0.073], [0.0147], [-3.121]]

Figure 21: FD_with_grid_5
Translation matrix : [[73.498], [182.699], [446.956]]
Rotation matrix : [[-0.00522], [-0.0212], [3.126]]

Figure 22: FD_with_grid_6
Translation matrix : [[58.473], [183.001], [437.982]]
Rotation matrix : [[0.145], [0.0143], [3.113]]

Figure 23: FD_with_grid_7
Translation matrix : [[50.427], [186.877], [428.537]]
Rotation matrix : [[0.247], [0.0403], [3.0825]]

Figure 24: FD_with_grid_8
Translation matrix : [[68.043], [181.545], [431.864]]
Rotation matrix : [[0.367], [0.0774], [3.042]]

6.3. Dataset HG:

Figure 25: HG_1

Figure 26: HG_2

Figure 27: HG_3

Figure 28: HG_4

Figure 29: HG_5

Figure 31: HG_7

Figure 30: HG_6

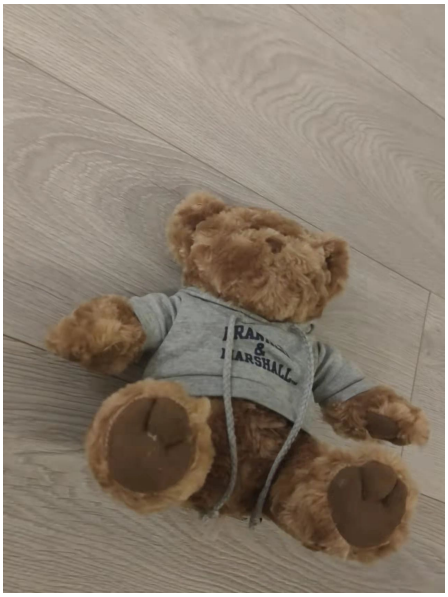

Figure 32: HG_8

6.4. Supplementary results:

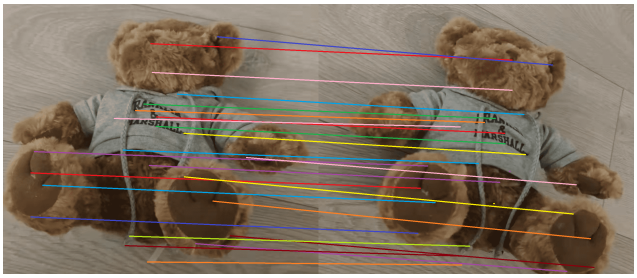

Figure 33: Manual Comparison between HG_3 and HG_8.

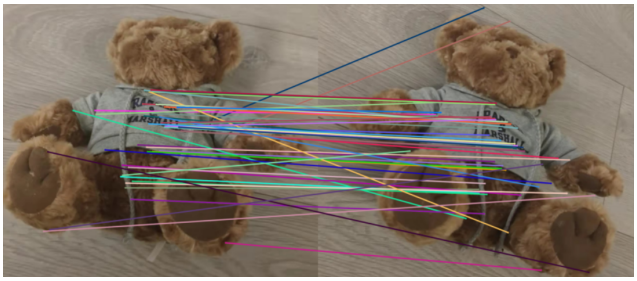

Figure 34: Automatic Comparison between HG_3 and HG_8. (top 50 matches plotted)

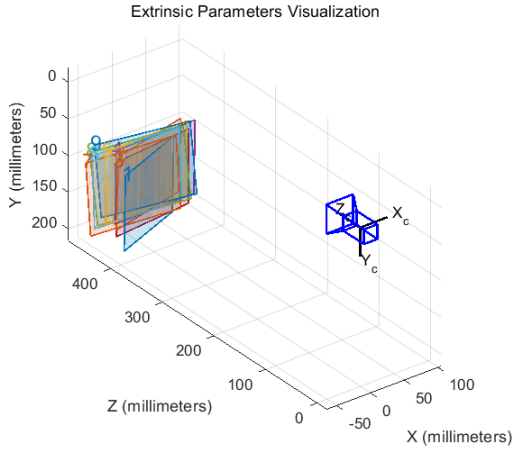

Figure 35: Extrinsic Parameters Visualization generated by MATLAB camera calibration tool.

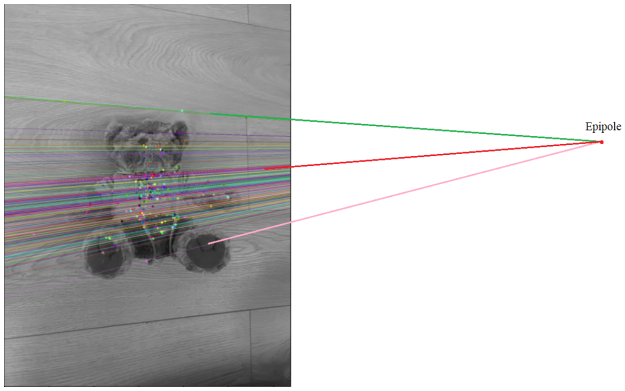

Figure 36: Epipole position of image FD_without_grid_5.

Figure 37: Stereo rectify result between FD_with_grid_4 and FD_with_grid_7 using the ”stereoCalibrate” and ”stereoRectify” function.