1. Data Collection

FD and HG images The FD set of images consists of 8 images taken of a calculator within a calibration grid from varying camera positions. The HG set of images consists of 9 images taken of the same calculator within the same calibration grid from a fixed camera position, but with different rotation angles of the camera and a zoom of 1.5x. 5 images of the empty grid are also taken for calibration purposes.

2. Keypoint correspondences

Correspondences found using a manual method Correspondences are manually clicked for a pair of images using the *Control Point Selection Tool* in MATLAB [1][2].

Correspondences found using an automatic method Figure 1, First, images were converted to grayscale to enhance the processing speed. The object used is a calculator with many buttons which allow for easy edge detection even in grayscale. Features in the image were detected using two methods: (i) the Harris–Stephens algorithm [3], and (ii) the detection of SURF features [4].

Descriptors and their locations are extracted from an image, and matched between a pair of images to obtain correspondence points. The Harris-Stephens algorithm detects corners, defined as the intersection of two edges, which makes it appropriate for the images used, since the calculator has many buttons. This makes it easier to identify regions of local intensity maxima or minima in the image. The hyperparameters used were tuned empirically based on visual inspection of the correspondences.

SURF is a local feature detector and its feature descriptor uses Haar wavelets as its basis function. It is able to detect features at different scales, and the descriptors are also rotation invariant. This makes it suitable for feature detection and matching in HG images, since they are taken from different rotation angles, and possibly at different zooms.

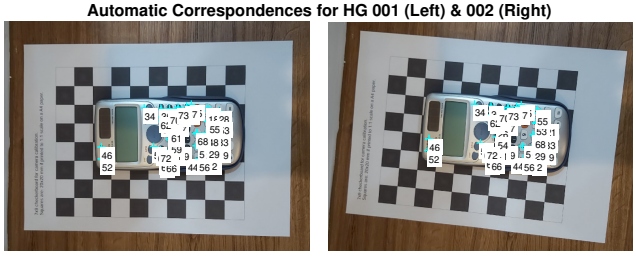

Figure 1: Correspondences found automatically through the matching of SURF features for a pair of HG images

Comparison between correspondences found by manual and automatic methods Table 1: Quantity and quality; Table 2: Cross validation error, Quality is defined as the percentage of correspondences that are determined to be correct through visual inspection, which implies that the manual method has a quality metric of 100%. In order to obtain a more objective metric, nested cross validation (CV) can be used to estimate the quality of correspondences (Appendix A). The CV error estimates on average, how many pixels away the prediction is from the labelled point.

Method	FD pair		HG pair	
	Quantity	Quality (%)	Quantity	Quality (%)
Manual	20	100	20	100
Harris	13	92.31	89	91.01
SURF	98	89.80	26	92.31

Table 1: Comparison between correspondences found using a manual and two automatic methods (Harris and SURF) for a pair of FD images and a pair of HG images

Method	FD pair	HG pair
Manual	3.32 ± 0.75	2.87 ± 0.10
Harris	6.53 ± 3.56	36.07 ± 7.89
SURF	60.09 ± 37.96	7.28 ± 0.78

Table 2: Cross validation errors for correspondences found using a manual and two automatic methods (Harris and SURF) for a pair of FD images and a pair of HG images

Unsurprisingly, the manual method of detecting correspondences led to the highest quality and the lowest CV error for all the trials. This suggests that the manual method of detecting correspondences is more accurate than the automatic methods, however, it can be tedious and time consuming which is why it has a relatively low quantity of correspondences labelled.

For this pair of FD images, the Harris algorithm resulted in a lower CV error than SURF feature matching which implies that the Harris algorithm performs better on images that are not rotated. On the other hand, for this pair of HG images, SURF feature matching resulted in a lower CV error than the Harris algorithm, suggesting that SURF feature matching performs better for images that are rotated, or have different zooms. Comparisons of the three methods on more FD and HG pairs are included in Appendix A.

3. Camera calibration

Camera parameters The *Camera Calibrator* in MATLAB uses correspondence points from multiple images to estimate the camera parameters [5].

$$\text{Intrinsic Matrix} = 10^3 \begin{bmatrix} 3.1655 & 0 & 0 \\ -0.0068 & 3.1541 & 0 \\ 1.9554 & 1.4741 & 0.0010 \end{bmatrix}$$

$$\text{Principal Point} = 10^3 \begin{bmatrix} 1.9554 & 1.4741 \end{bmatrix}$$
$$\text{Focal Length} = 10^3 \begin{bmatrix} 3.1655 & 3.1541 \end{bmatrix}$$
$$\text{Skew} = -6.7696$$

Camera distortions Figure 2, The radial and tangential distortions of the camera are also estimated using the *Camera Calibrator* in MATLAB (Appendix B). The distortion coefficients are reported below.

$$\text{Radial Distortion} = \begin{bmatrix} 0.2223 & -0.8254 \end{bmatrix}$$
$$\text{Tangential Distortion} = \begin{bmatrix} -0.0046 & -0.0083 \end{bmatrix}$$

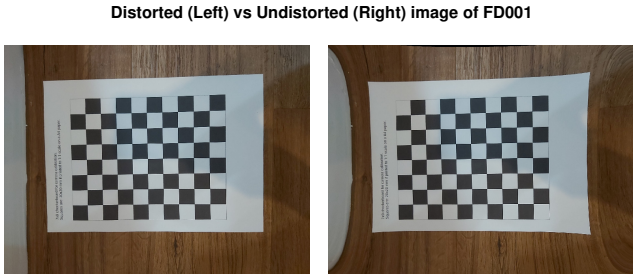

Figure 2: Effects of radial and tangential distortions on an FD image

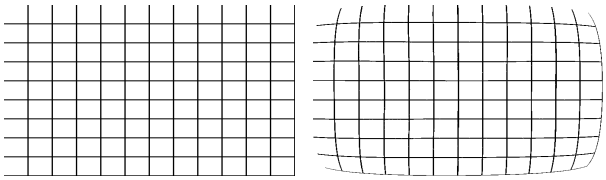

Figure 3: Left: Original virtual grid, Right: Virtual grid following radial and tangential distortion

The presence of radial distortion means that pixels are non-square (Figure 3, right). The distortion gets larger for coordinates further away from the centre of the image.

4. Transformation estimation

Homography matrix The homography matrix is calculated using singular value decomposition (svd) in MATLAB [6]. The Hough transform and RANSAC are not required because there are no outliers in the correspondences. Four correspondences are required to solve for the homography matrix [7], however, more than 4 correspondences were found which means that the system of equations is overdetermined. To overcome this, the svd function implements ordinary least squares and returns the homography matrix which minimises the sum of square residuals. The homography matrix H between the pair of images HG003 and HG004 is reported below.

$$H = \begin{bmatrix} 0.0014 & 0.0003 & -0.3698 \\ -0.0003 & 0.0014 & 0.9291 \\ 0 & 0 & 0.0015 \end{bmatrix}$$

The term H_{33} is not 1 since the matrix is normalised such that the sum of squared parameters equals to 1 (i.e. $H_{11}^2 + H_{12}^2 + ... + H_{33}^2 = 1$). The terms H_{31} and H_{32} are 0 which implies that the transformation is an affine transformation, a linear transformation. The top left 2×2 sub-matrix in H accounts for scaling and rotation, and the top right 2×1 sub-matrix accounts for translation [8]. Since H_{11} is positive, and is associated with $\cos(\alpha)$, it can be deduced that the rotation angle is positive and acute which is also seen in Figure 4.

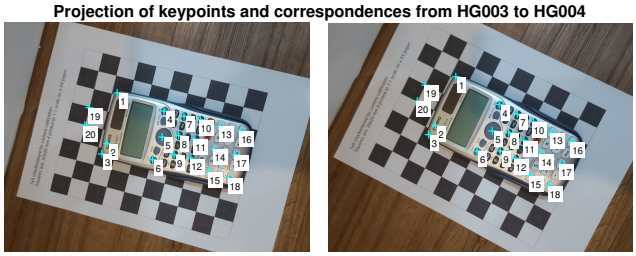

Figure 4: Projection of keypoints and correspondences from HG003 to HG004. A good estimate of the homography matrix was obtained since the correspondence points are maintained between the two images.

Fundamental matrix The MATLAB function *estimateFundamentalMatrix()* uses the normalised eight-point algorithm to estimate the fundamental matrix from corresponding points in a pair of stereo images [9], which is reported below for the image pair FD003 and FD004. The Hough transform and RANSAC are not required because there are no outliers in the correspondences. Since epipoles lie on epipolar lines, F cannot be full rank, and instead it has a rank of 2. This is why the middle row of the fundamental matrix below is zero. Furthermore, the left and right kernels of the fundamental matrix define the epipoles [10].

For the images shown in Figure 5, the epipole is the point of intersection of the epipolar lines. Vanishing points can also be found by extending lines that are parallel in the 3D world, and locating the point of intersection. The horizon then passes through the vanishing point.

$$F = \begin{bmatrix} 0 & 0 & -0.0020 \\ 0 & 0 & 0 \\ 0.0015 & -0.0007 & 1 \end{bmatrix}$$

$$\mathrm{e_r} = \begin{bmatrix} 0.0152 & 0.9999 & 0.0007 \end{bmatrix}$$
$$\mathrm{e_l} = \begin{bmatrix} 0.3291 & 0.9443 & 0.0007 \end{bmatrix}$$

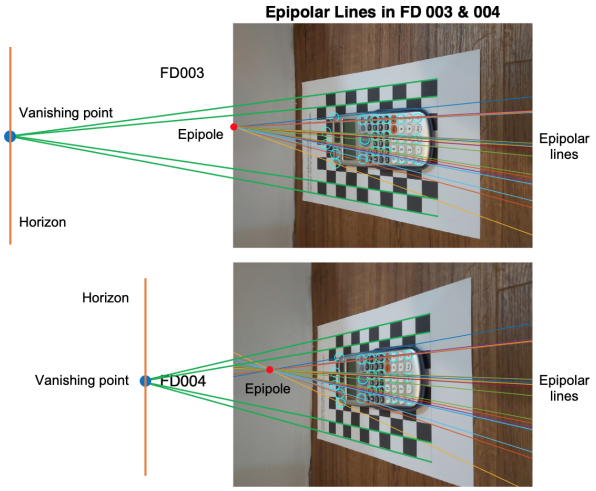

Figure 5: Keypoints and correspondences between a pair of FD images with epipolar lines (multi-coloured), epipoles (red), vanishing points (blue), and the horizon (orange lines)

5. 3D geometry

Stereo rectified images Figure 6, The MATLAB function *rectifyStereoImages()* was used to stereo rectify the image pair FD001 and FD002 such that epipolar lines are all horizontal, and correspondences lie on the same vertical level [11][12].

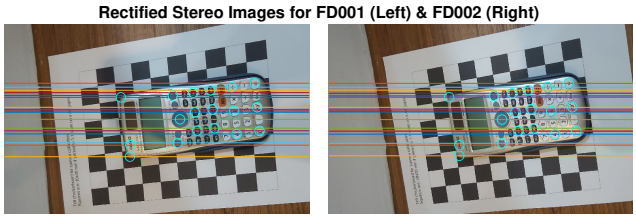

Figure 6: Stereo rectified pair of FD images with horizontal epipolar lines

Depth map Figure 7, The MATLAB function *disparityBM()* uses block matching to compute a disparity map [13]. The block matching algorithm splits the image into macroblocks to search for correspondences between the stereo rectified image pair. Based on the difference in position of the same 3D world point on the two images (disparity), an estimate of depth can be obtained. A larger difference between the position of a 3D point on the two images reflects a shallower depth [14].

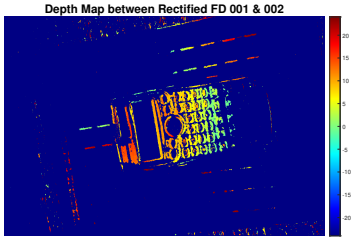

Figure 7: Depth map of the scene. A dark blue colour represents 3D points further away from the camera, and dark red represents 3D points closer to the camera.

The estimate for the depth of the right side of the calculator is not accurate since it appears as blue (further away from camera) in Figure 7. One possible reason is because of different lighting conditions that fell on the right side of the calculator when the images were taken. Due to this, few correspondences in the affected area were detected, hence, the estimate of depth in this region suffered.

6. Conclusions

Correspondences between pairs of images were detected and matched successfully, with the Harris algorithm being more suited for FD images, and SURF feature matching being more suited for HG images. The correspondences allowed for the estimation of the homography matrix which can be used to project points from one image onto another. Furthermore, estimation of the fundamental matrix revealed the epipoles, and also allowed for stereo rectification of images. Finally, a depth map was also produced from a pair of stereo images.

References

[1] The Mathworks, Inc., Natick, Massachusetts, *MATLAB version 9.7.0.1296695 (R2019b) Update 4*, 2019.

[2] "Mathworks matlab control point selection tool." https://www.mathworks.com/help/images/ref/cpselect.html. Accessed: 2021-02-12.

[3] "Mathworks matlab detect corners using harris–stephens algorithm." https://www.mathworks.com/help/vision/ref/detectharrisfeatures.html. Accessed: 2021-02-12.

[4] "Mathworks matlab detect surf features." https://www.mathworks.com/help/vision/ref/detectsurffeatures.html. Accessed: 2021-02-12.

[5] "Mathworks matlab single camera calibrator to estimate camera intrinsics, extrinsics, and lens distortion parameters." https://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html. Accessed: 2021-02-12.

[6] "Mathworks matlab singular value decomposition." https://www.mathworks.com/help/matlab/ref/double.svd.html. Accessed: 2021-02-12.

[7] K. Mikolajczyk, "Computer vision and pattern recognition lecture 3, slide 15 estimating homography matrix."

[8] K. Mikolajczyk, "Computer vision and pattern recognition lecture 2, slide 12 affine transformation."

[9] "Mathworks matlab estimate fundamental matrix from corresponding points in stereo images." https://www.mathworks.com/help/vision/ref/estimatefundamentalmatrix.html. Accessed: 2021-02-12.

[10] K. Mikolajczyk, "Computer vision and pattern recognition lecture 3, slide 22 epipolar geometry."

[11] "Mathworks matlab rectify a pair of stereo images." https://www.mathworks.com/help/vision/ref/rectifystereoimages.html. Accessed: 2021-02-12.

[12] K. Mikolajczyk, "Computer vision and pattern recognition lecture 4, slide 25 stereo rectification."

[13] "Mathworks matlab compute disparity map using block matching." https://www.mathworks.com/help/vision/ref/disparitybm.html. Accessed: 2021-02-12.

[14] K. Mikolajczyk, "Computer vision and pattern recognition lecture 4, slide 17 relationship between disparity and depth."

Appendix

A. Nested Cross Validation Method

In order to obtain a more objective metric, nested cross validation (CV) can be used to estimate the quality of correspondences, assuming that the mean transformation matrix approaches the true matrix as the number of correspondence points increases. First, the data set was split into k-folds where $k = 3$. At each outer CV loop, one fold is used as the testing set and all the other folds are used as the training set. Next, leave one out CV was performed on the training set in order to select the transformation matrix which led to the lowest inner CV error, defined as the mean L_2 distance between the predicted coordinates and the labelled coordinates. Finally, this transformation matrix was used on the testing set to determine the CV error which indicates on average, how many pixels away the prediction is from the labelled point. One limitation was the small sample size. As a result, the variance of CV error across multiple pairs is large.

Method	FD pair 1	FD pair 2
Manual	3.32 ± 0.75	3.10 ± 0.72
Harris	6.53 ± 3.56	3.01 ± 1.34
SURF	60.09 ± 37.96	27.73 ± 8.26

Table 3: Cross validation error for correspondences found using manual and automatic methods (Harris and SURF) for two pairs of FD images

Method	HG pair 1	HG pair 2
Manual	2.19 ± 0.26	2.87 ± 0.10
Harris	40.59 ± 16.25	36.07 ± 7.89
SURF	48.74 ± 36.01	7.28 ± 0.78

Table 4: Cross validation error for correspondences found using manual and automatic methods (Harris and SURF) for two pairs of HG images

B. Radial and tangential distortions

Radial distortion occurs when light rays bend more near the edges of a lens than they do at the optical center.

$$\text{Radial Distortion} = \begin{bmatrix} 0.2223 & -0.8254 \end{bmatrix}$$

The elements in the radial distortion matrix correspond to the radial distortion coefficients of the lens k_1 and k_2. The mapping between the original and distorted coordinates is

then given by:

$$x_{\text{distorted}} = x\left(1 + k_1 r^2 + k_2 r^4\right)$$
$$y_{\text{distorted}} = y\left(1 + k_1 r^2 + k_2 r^4\right)$$

where x and y correspond to the original coordinates, $x_{\text{distorted}}$ and $y_{\text{distorted}}$ correspond to the distorted coordinates, and $r^2 = x^2 + y^2$.

Tangential distortion occurs when the lens and the image plane are not parallel.

$$\text{Tangential Distortion} = \begin{bmatrix} -0.0046 & -0.0083 \end{bmatrix}$$

The elements in the tangential distortion matrix correspond to the tangential distortion coefficients p_1 and p_2. The mapping between the original and distorted coordinates is then given by:

$$x_{\text{distorted}} = x + \left(2p_1 xy + p_2(r^2 + 2x^2)\right)$$
$$y_{\text{distorted}} = y + \left(2p_2 xy + p_1(r^2 + 2y^2)\right)$$

where x and y correspond to the original coordinates, $x_{\text{distorted}}$ and $y_{\text{distorted}}$ correspond to the distorted coordinates, and $r^2 = x^2 + y^2$.