Computer Vision Coursework Report – Leili Barekatain

Question 1

I used **SIFT** (Scale-Invariant Feature Transform) because I wanted to identify distinctive keypoints that are rotation-invariant, scale-invariant, and robust to illumination changes (using gradient information). SIFT focuses on **local image features, like edges and corners**, which are robust and distinctive enough to be matched across frames. This ensures consistent and repeatable features, which are crucial for estimating the fundamental matrix. Additionally, SIFT detects keypoints with precise localization. It is a robust, distinctive, compact and efficient descriptor. SIFT operates in four steps:

1. **Scale-space Extrema Detection**: Detects keypoints by identifying areas of high contrast in a scale space created using Gaussian filters
2. **Keypoint Localization**: Refining candidate keypoints by removing weak, noisy, or edge responses
3. **Orientation Estimation**: Calculating an orientation for each keypoint relative to the dominant orientation in its neighbourhood
4. **Keypoint Descriptor**: Generating a unique descriptor for each keypoint that summarizes the gradient patterns in its local area

Question 2

To match the detected salient features between video frames, I used **Nearest Neighbor Distance Ratio.** It estimates the distance of a feature vector to its nearest neighbor (d1) and its second nearest neighbor (d2). If $d1 \approx d2$, the matching is ambiguous and should be rejected and if $d1 << d2$ the matching is correct. Steps are as follows:

1. **Feature Extraction**: Detect and describe features in each frame using SIFT
2. **NN Matching**: Use the NN algorithm to find the two nearest matches in frame2 for each descriptor in frame1 based on Euclidean distance
3. **Ratio Test**: Apply the ratio test to ensure the closest match distance is significantly smaller than the second-closest one. A high threshold (close to 1) in the ratio test allows more, but potentially ambiguous, matches to pass. A low threshold (around 0.4 or 0.5) makes the test stricter, allowing only confident matches to pass.

Question 3

a) Plot the detected features on the provided pair of frames (using SIFT)

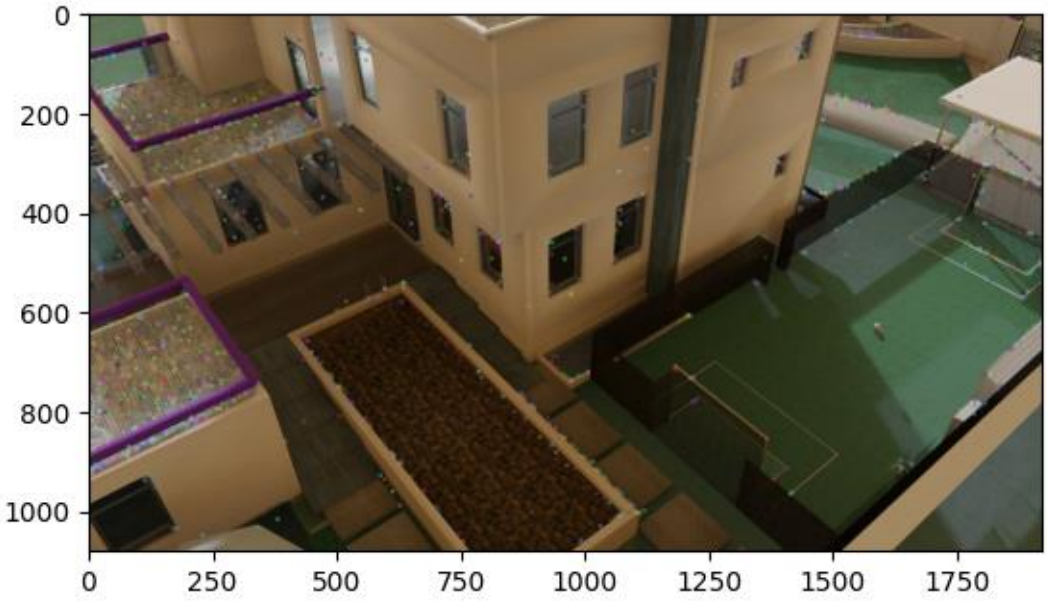

Figure 1: Salient Features Detected in Frame 1

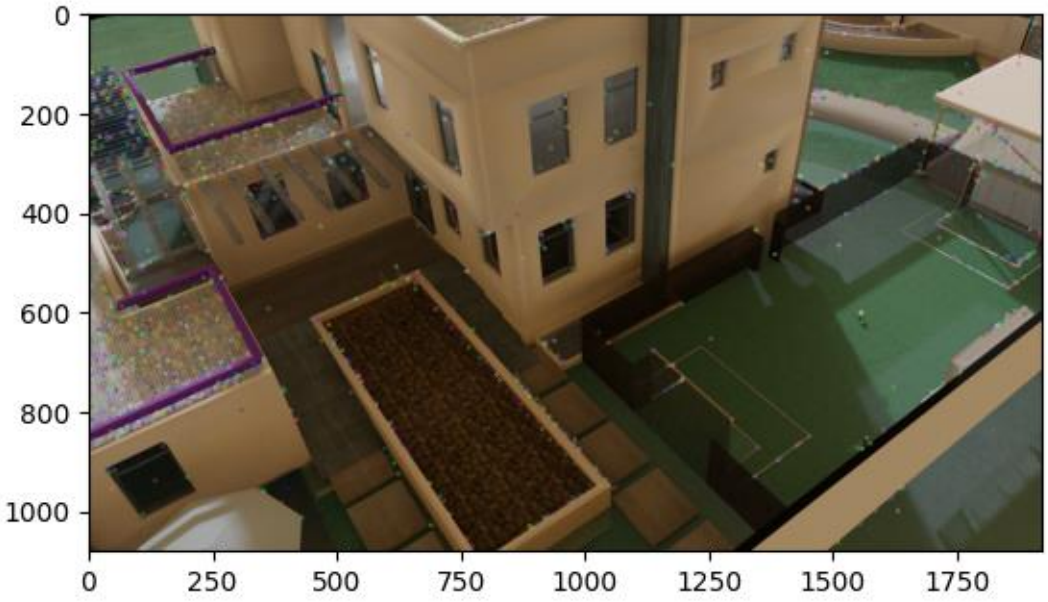

Figure 2: Salient Features Detected in Frame 2

b) Illustrate matches between two frames (found using NN)

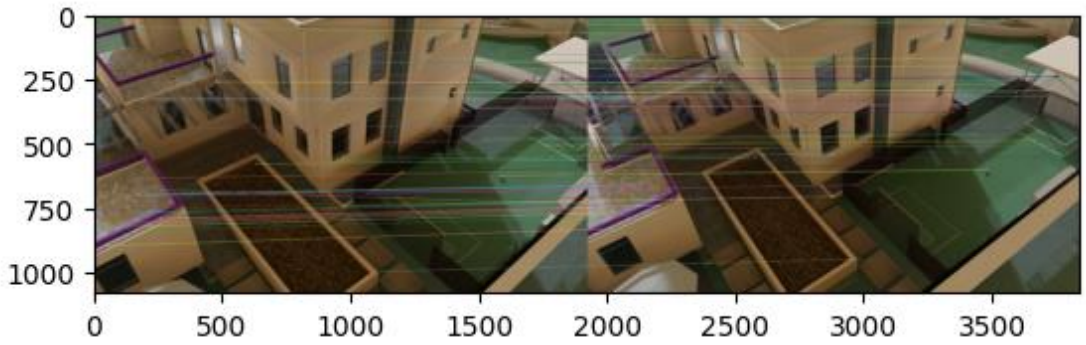

Figure 3: Feature Matches Between Frames (ratio=0.4) – version 1

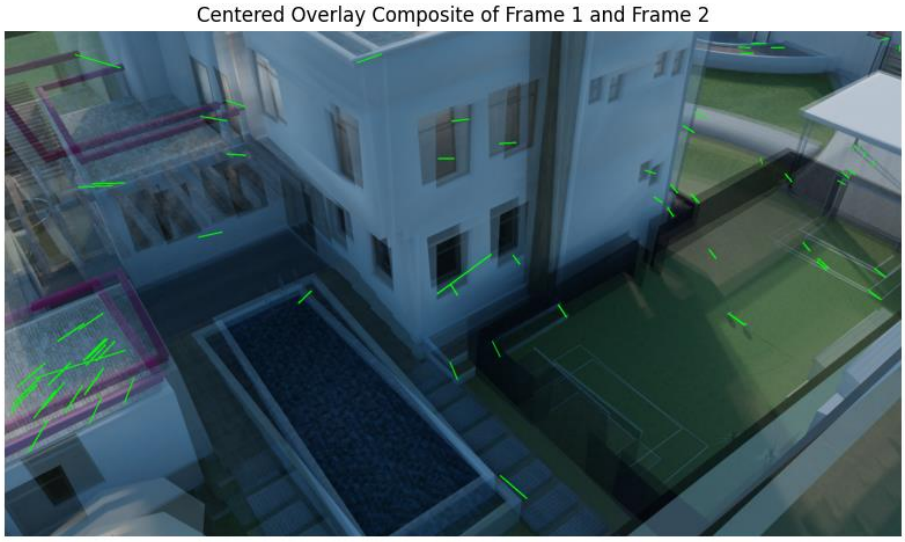

Figure 4: Feature Matches Between Frames (ratio=0.4) – version 2

C) Compare the estimated fundamental matrices and explain any possible disagreement between the two methods. Which method is more accurate? Justify your answer and suggest how you could improve the least accurate method.

The results are as follows:

```
Fundamental matrix from matched features:
 [[ 3.23185408e-07 -1.66433797e-05  4.35229840e-04]
 [ 1.77033297e-05 -1.49169715e-06 -3.01318629e-02]
 [-1.58861904e-03  2.88683561e-02  1.00000000e+00]]
```

```
Fundamental matrix from camera parameters:
 [[-4.01210372e-10 -8.78019327e-08  4.97699667e-05]
 [ 2.51186447e-08  1.36583164e-09  1.31446124e-03]
 [-1.88336735e-05 -1.15844731e-03 -5.24927926e-02]]
```

To evaluate the accuracy of the fundamental matrix, we can check if it meets the Epipolar Constraint, meaning it should satisfy the condition $X'^{T}FX = 0$ for corresponding points x and x′ in two frames. The closer this value is to zero, the more accurately the matrix meets the Epipolar Constraint.

I calculated the Epipolar Constraint Error for both fundamental matrices:

```
Average Epipolar Constraint Error for Estimated F based on matched features: 0.051244503830011424
Average Epipolar Constraint Error for Estimated F based on camera parameters: 0.006113334969635735
```

The fundamental matrix from camera parameters is more accurate due to intrinsic and extrinsic factors. However, estimating with the matched features can be less reliable due to mismatched points or noise in feature detection.

To improve the accuracy of the matched features method, we can filter more matches by decreasing the NNDR ratio. Also, we can use a more robust matching algorithm (e.g., RANSAC) to reduce the impact of outliers. Refining feature matching with techniques like Bundle Adjustment can also enhance alignment.

D) Illustrate correctly matched points that meet the Epipolar constraint. Briefly explain how these matches have been identified.

To identify correctly matched points that satisfy the Epipolar constraint, I used the following approach:

1. **Compute Epipolar Constraint**: For each pair of matched points, I calculated the Epipolar constraint $X'^{T}FX = 0$, where F is the fundamental matrix computed using camera parameters (the more accurate one) and x and x' are the points in the two images.
2. **Apply a Threshold**: I checked if the computed constraint value is close to zero (within a small threshold of 0.01), indicating that the points lie on each other's epipolar lines and are therefore geometrically consistent.
3. **Filter Matches**: Matches that meet the Epipolar constraint within the specified threshold are retained as correct matches, while others are discarded.
4. **Illustrate Matches**: I visualized the matches in two ways which are shown below.

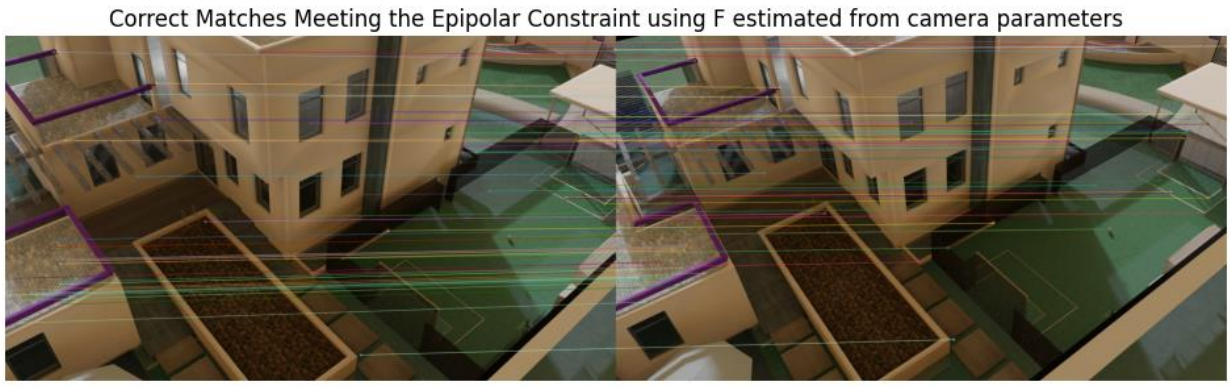

Figure 5: Correctly matched points that meet epipolar constraint – version 1

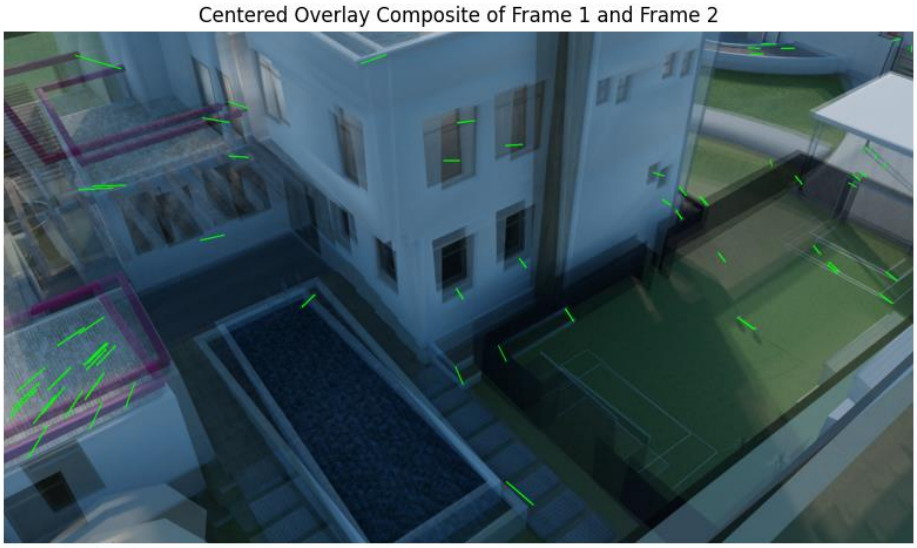

Figure 6: Correctly matched points that meet epipolar constraint – version 2

e) Estimate the area of the swimming pool and the length (touchline) of the football field

In order to calculate the area of the pool and the length of the field, the steps I followed are as follows:

1. **Rectification:** I rectified both frames and found the new baseline and focal length from the Q matrix (disparity-to-depth mapping matrix), using this formula:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{\text{baseline}} & 0 \end{bmatrix}$$

2. **Identifying Points Manually**: I identified the coordinates of 3 points (Top-left, Top-right, and Bottom-right) of the pool and 2 points (Top-left and Bottom-right) of the field manually in both frames.

3. **Disparity Calculation for Depth**: For each of these matching points in two frames (x and x'), I calculated the disparity (difference in the x-coordinates between the two images) to derive the depth, using this formula:

$$\text{depth} = \frac{\text{baseline} \times \text{focal\_length}}{x - x'}$$

4. **Transformation to 3D Coordinates**: I converted each 2D point from the image space into 3D coordinates in the camera's coordinate system (reference system), using this formula:

$$X = (x - c_x) \cdot \frac{\text{depth}}{\text{focal\_length}}, \quad Y = (y - c_y) \cdot \frac{\text{depth}}{\text{focal\_length}}, \quad Z = \text{depth}$$

5. **Calculating Side Lengths:** I used the 3D coordinates to calculate the lengths of the two adjacent sides (Top-left to Top-right and Top-right to Bottom-right) of the pool and the length of one side of the field in 3D space. These distances represent the real-world lengths of the pool's width and height and the field's touchline:

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Finally, I multiplied the two side lengths to estimate the area of the pool. These are my results:

Area of the pool: 29.4

Length of the field: 15.26

Optional: Illustrate the disparity map and the rectification result for the above video frames

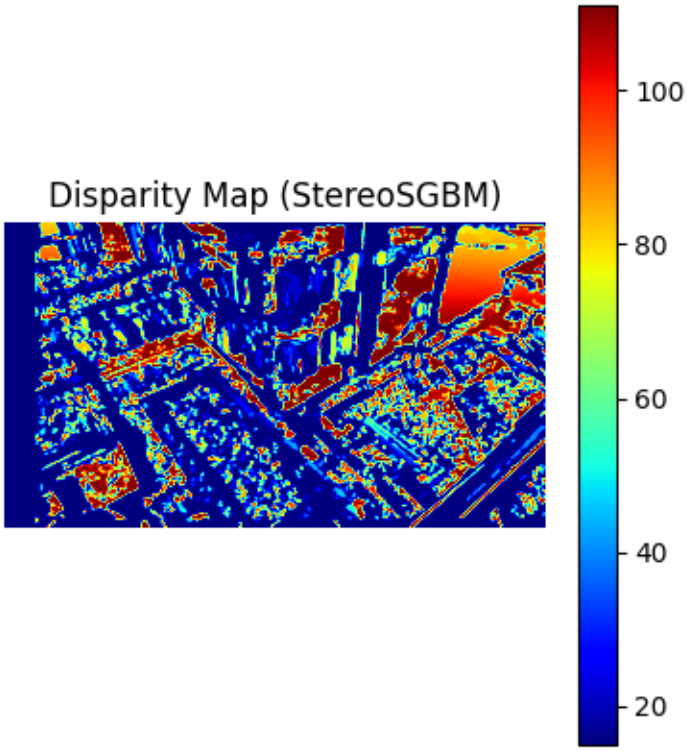

Figure 7: Disparity Map

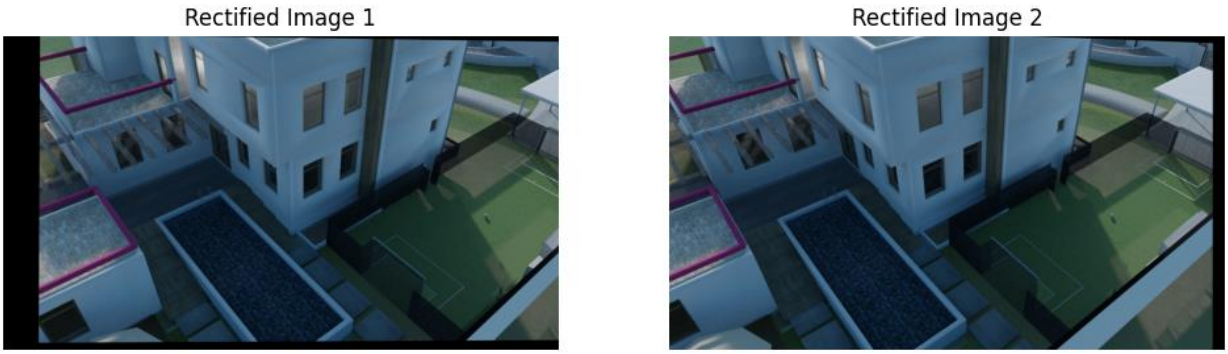

Figure 8: Results of applying rectification