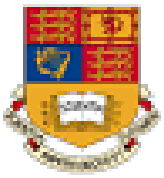


Imperial College London

COMPUTER VISION

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

FEATURE EXTRACTION AND SURFACE RECONSTRUCTION

Author:
Joshan Dooki

CID No:
02182106

Course Lead:
Stamatia (Matina) Giannarou, PhD

November 20, 2022

1 Propose a technique to detect salient features of your choice on the video frames above. Explain the type of features on which you will focus and justify your choice. (Word limit: 150 words)

Based on the images provided, the features considered were (1) edges, (2) corners, (3) blobs and (4) ridges. There will be a stronger focus on edges and corners since the images provided have well defined straight lines. Local features are preferred because it allows for extraction to be more generalizable which in turn improves efficiency.

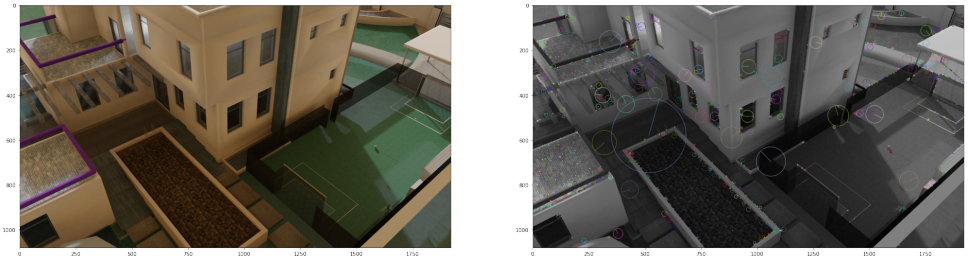
From the several algorithms considered, the SIFT method was the chosen . The Harris and Shi-Tomasi corner detectors were considered for extracting corners, the Canny edge detector was considered for edge-detection, and LoG, DoG and DoH was considered for blob detection[1]. Each of these methods however, fails when there is image rotations or size changes. Hence, the Sift algorithm was chosen since it is able to detect salient features in both images regardless of orientation and size changes. SURF is a faster version of SIFT but was not accessible due to patent restrictions.

2 Propose a technique to match the detected salient features between the video frames. Explain how you would approach this task and the steps you would follow. (Word limit: 150 words)

In order to perform stereo rectification , salient features between each image needs to be matched. The Flann matcher was chosen since this algorithm sorts the best potential matches between similar key points in frame 1 and frame 2. This is based on the distance using a KNN search criteria, which is a variant of the nearest neighbours technique detailed by Lowe [3]. The alternative algorithm considered was the brute force matcher. Flann was still preferred since it works faster, its collection of algorithms work best for nearest neighbors searches and it has the ability to choose the best algorithm and optimize parameters[2].The steps followed for feature matching are:
1. Feature Detection: (explained in Q1)
2. Description: The method implemented was SIFT which produces vector descriptors.
3. Feature matching: The descriptors are compared across images to identify similar features. Using Flann, the KNN parameter, K, was set at 2 and the ratio of distances (closest/next closest) was set to 0.8. This value was chosen since according to D.Lowe, this threshold eliminates 90% of false matches and only 5% of correct matches [4].

3 Use a programming environment of your choice to:

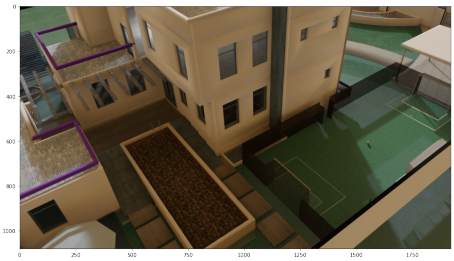
3.1 a. Implement your proposed salient feature detector and plot the detected features on the provided pair of frames.



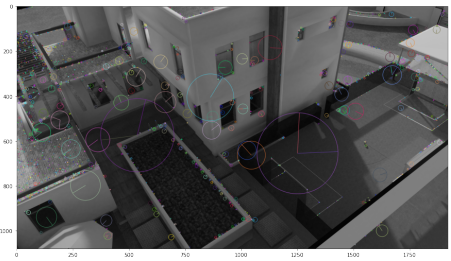
(a) Original image (b) Features detected using SIFT

Figure 1: Feature Extraction- Frame 1

N.B. Images have been converted to grey scale to reduce computational cost of feature detection



(a) Original image



(b) Features detected using SIFT

Figure 2: Feature Extraction- Frame 2

3.2 b. Find corresponding features between the two frames and illustrate those matches. To illustrate the matches you can, for example, create a composite image (e.g centered overlay image) from the two frames.

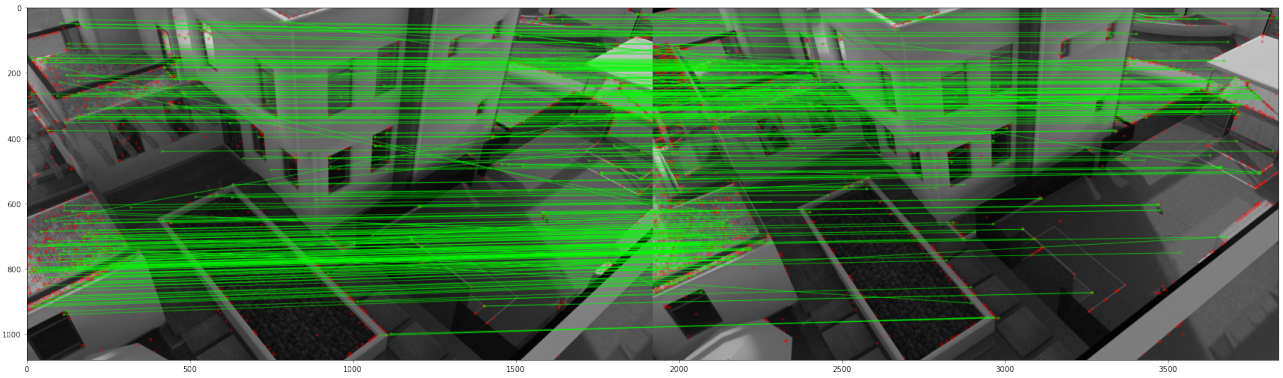


Figure 3: Matched features using SIFT

The flann algorithm found a total of 1469 matches between frame 1 and frame 2

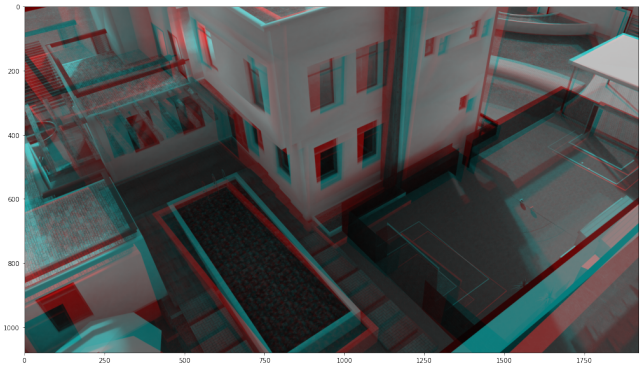


Figure 4: Composite image

3.3 c. Use the matched features to estimate the fundamental matrix between the two images. Now estimate the fundamental matrix using the extrinsic and intrinsic camera parameters. 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. (Word limit: 150 words)

Estimated fundamental matrix using cv.findFundamentalMat

$$F^{(estimated)} = \begin{bmatrix} -1.098e-07 & 7.510e-07 & 1.874e-03 \\ 1.0288e-06 & -2.020e-07 & -3.498e-02 \\ -2.337e-03 & 3.127e-02 & 1.000e+00 \end{bmatrix}$$

Estimated fundamental matrix using extrinsic and intrinsic parameters

$$F^{(calculated)} = \begin{bmatrix} 7.634e-09 & 1.673e-05 & -9.481e-02 \\ -4.785e-06 & -2.602e-07 & -2.504e-01 \\ 3.588e-03 & 2.207e-01 & 1.000e+00 \end{bmatrix}$$

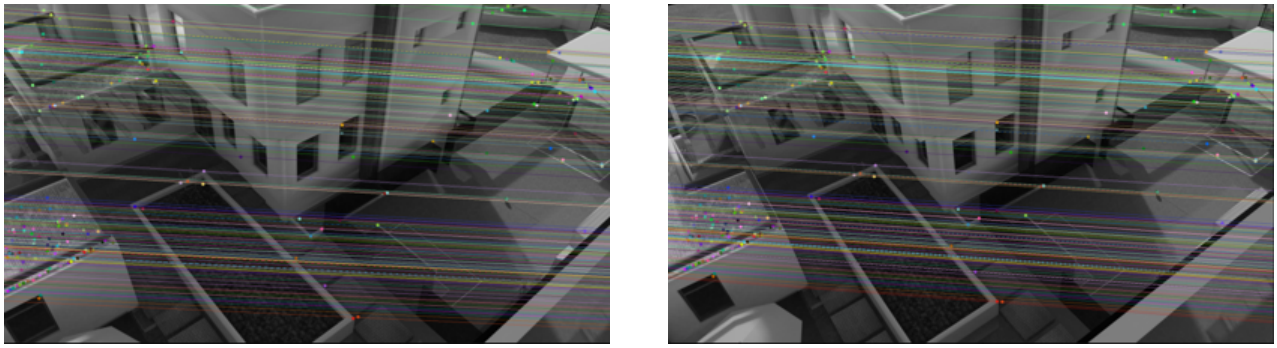
The following formula was applied to calculate the fundamental matrix

$$\begin{aligned} F &= e' * [P'.P^{+}] \\ &= [P'.C] * [P'.P^{+}] \\ &= [K'.t]x * K.R.K^{-1} \end{aligned}$$

where K' is the intrinsic matrix of camera 2, K is the intrinsic matrix of camera 1, R is rotation matrix and t is transformation matrix.

Both $F^{(calculated)}$ and $F^{(estimated)}$ are of relatively similar magnitudes. Hence, they can be considered insignificantly different. The difference between these two F matrices are due to $F^{(estimated)}$ being calculated using estimated key point matches using Flann. The $F^{(calculated)}$ is more accurate than $F^{(estimated)}$ since $F^{(estimated)}$ is obtained based on the key point matches calculated using the Flann algorithm. $F^{(estimated)}$ can be improved by using other feature matching algorithms (such as brute force and ORB) and conducting hyper-parameter tuning on the Flann-algorthim

3.4 d. Find the correctly matched points that meet the epipolar constraint and illustrate these matches. Briefly explain how these matches have been identified. (Word limit: 150 words)



(a) Frame 1 matched points

(b) Frame 2 matched points

Figure 5: Feature matching

Since feature matching has been complete using Flann (in Q3b), every point 'x' observed by camera 1 must be observed along camera 2's epipole lines. This produces the epipole constraint, where the projection of key points in frame 1 must be contained in the epipole lines of frame 2. These epipole constraints are required for stereo rectification. Stereo rectification re-projects the image to a new common plane which is parallel to the lines between the camera centers. The fundamental matrix was used to calculate the epipole lines in the second image. Open cv's cv.findFundamentalMat() function was used to calculate the fundamental matrix based on matched key point pairs. Once the fundamental matrix was obtained, the epipole lines were produced which then visually represents the key point matches on both frames.

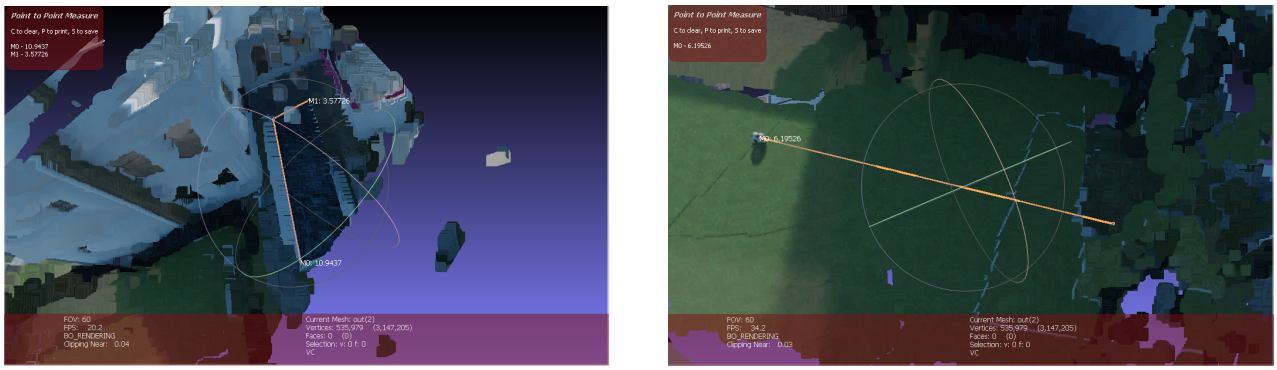
3.5 e. Estimate the area of the swimming pool and the length (touchline) of the football field. (hint: you can establish the disparity map between these frames or you can apply 3D surface reconstruction)

The rectification transforms for each head of the calibrated stereo cameras was computed using cv.StereoRectify. This produces the Q value which is a 4 X 4 disparity-to-dept mapping matrix. This Q value is then used in conjunction with the disparity map (created using cv.StereoMatcher.compute) as inputs to creating the 3D images. The function cv.reprojectImageTo3D was used to create the 3D image. This function transforms the single channel disparity map to a 3-channel image representing a 3D surface. For each pixel(x,y) and corresponding disparity, it computes:

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = Q \begin{bmatrix} X \\ Y \\ disparity(x,y) \\ Z \end{bmatrix}$$

Both the area of the pool and length of the football field was estimated by conducting 3D surface reconstruction on the disparity map. Meshlab was the tool used to analyze the 3D image.
The estimated dimensions of the pool are 10.9m by 3.6m
* Area of pool $\approx$ 40m2

The length from the touchline to the center of the football pitch is 6.2m hence the total length is 2 X 6.2m
* Length of football field $\approx$12m



(a) 3D surface reconstruction of pool

(b) 3D surface reconstruction of Football pitch

Figure 6: 3D surface reconstruction

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

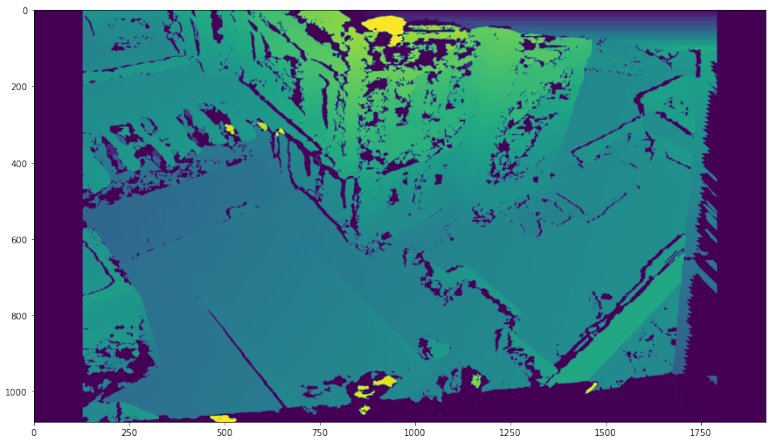
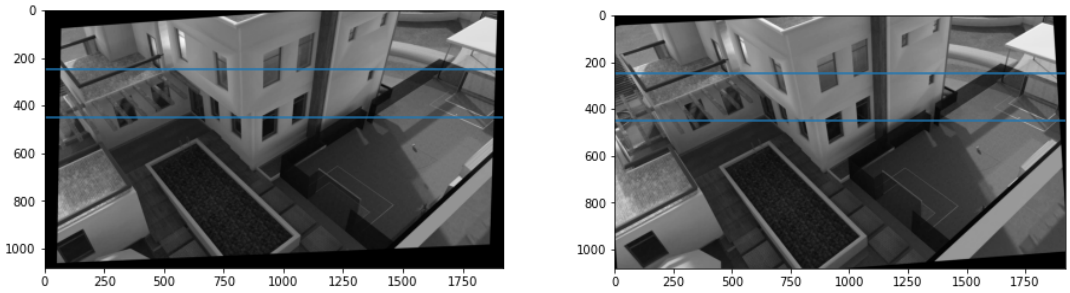


Figure 7: Disparity map



(a) Frame 1 rectified

(b) Frame 2 rectified

Figure 8: Rectified Frames

5 References

[1]Tuytelaars, T. and Mikolajczyk, K. (2007). Local Invariant Feature Detectors: A Survey. Foundations and Trends® in Computer Graphics and Vision, 3(3), pp.177–280. doi:10.1561/0600000017.

[2]Ailon, N. and Chazelle, B. (2009). The Fast Johnson–Lindenstrauss Transform and Approximate Nearest Neighbors. SIAM Journal on Computing, 39(1), pp.302–322. doi:10.1137/060673096.

[3]Perera, S.A. (2018). A Comparison of SIFT , SURF and ORB. [online] Medium. Available at: https://medium.com/@she comparison-of-sift-surf-and-orb-333d64bcaaea.

[4]Lowe, D.G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2), pp.91–110. doi:10.1023/b:visi.0000029664.99615.94.