CS436/5310/EE513 Computer Vision Fundamentals

Instructor's Name: Murtaza Taj

19 October 2020

Multi-camera System

Mini-Project 1

Requirements

1. 3 or more smartphones or IP webcams
2. Internet
3. Github account: You are required to maintain your code using Github repository

The overall idea of the project is to combine visual recognition and multi-camera geometry and build a useful application. You can chose an application of your choices it can be a home surveillance system, shop foot fall analytics, match coverage or score overlay system, wild life monitoring, multi-camera system for cars etc. Irrespective of the application area you choice you have to perform the following 6 tasks and deliver them at the mentioned deadline.

Project Proposal

You can complete the project in a group of 2 students

You must form the group within this week and submit your proposal by mid-night Fri 23rd Oct 2020 via following Google Form link:

https://forms.gle/UDPeaXcwz6euQUSE8

Task 1: Multi-camera Setup & Recording

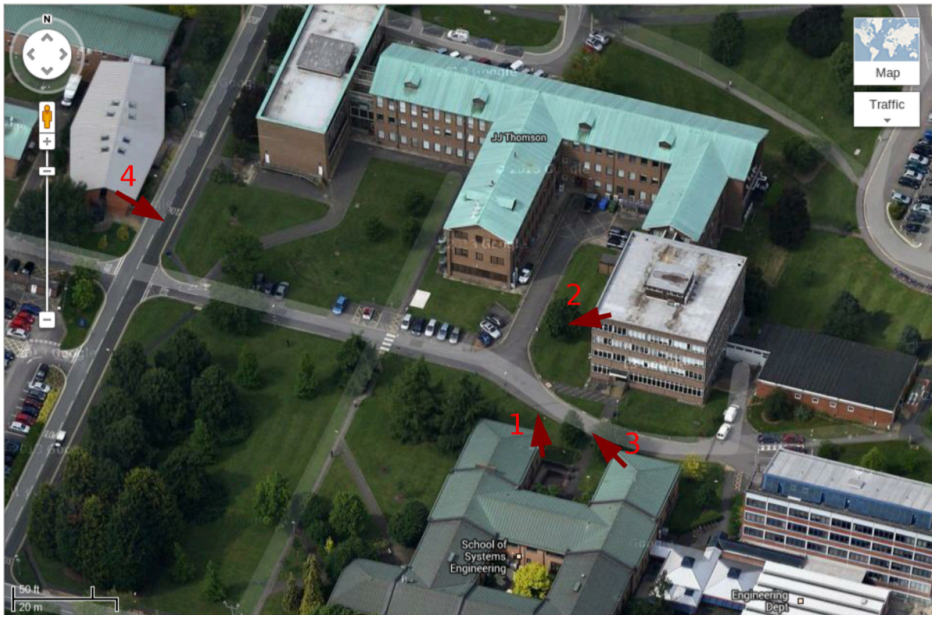
1.1 Collect three cameras within your home (smart phones, laptop camera etc) or from your friends, project partners etc. In the worse case scenario you can search for a multi-camera dataset on the web and use it for the project. But even in that case you should complete the code module for the recording setup.

1.2 Install *IP Webcam* app (or other similar app) on each of the smart phone

1.3 Open camera on smart phones and test the video feed on your laptop using *VLC* player

1.4 Write a program using python and *OpenCV* to record the feed from all these cameras on your hard drive. The program should also optionally display the live feed from all three cameras.

1.5 Once the recording setup is ready and tested mount the cameras at three different locations (for e.g. see image below)



Task 2: Object Detection (Person or Car)

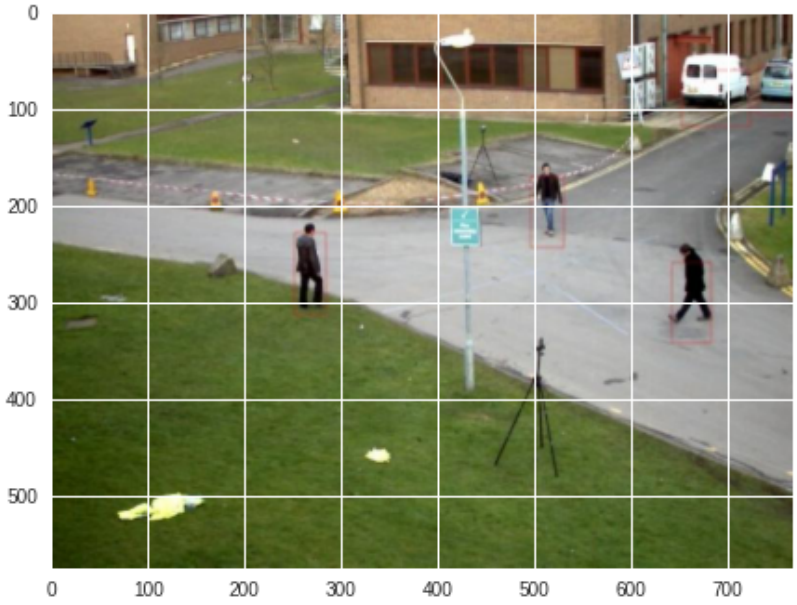
2.1 Setup and run YOLO object detector

2.2 From the recorded videos annotate few objects from each of the camera view using LabelMe software

2.3 Fine tune the YOLO object detector using the generated annotations. Depending upon your data you may fine tune *n* different versions of YOLO one for each camera view.

2.4 Write a program that takes perform object detection on the video feed obtained in Task 1. The program should have two modes online and offline. In online mode the program should read a frame from the live feed from each camera and perform object detection using YOLO object detector(s) and display the output from all the cameras simultaneously on the screen as well as record the bounding box position on the hard drive

2.5 Offline mode: In this task your are required to repeat the previous task for pre-recorded videos



Task 3: Orthographics Top-View Generation

3.1 Either using Google maps or otherwise obtain a top-view of the site

3.2 Mark (either manually or using GPS coordinates) the position of each camera on the top-view.

3.3 Find the corresponding points between top-view image and your camera image from first camera

3.4 Compute the homography between the corresponding points

3.5 Project the camera view on the top-view using the computed homography and produce a top-view video of the camera view (this option should work for both online and offline modes discussed earlier i.e. live feed and pre-recorded videos respectively)

3.6 Repeat the above process for all the cameras

3.7 Combine all the top-view videos into a single top-view video of the site. In this task a frame from each of the camera will be obtained and projected on the common top-view and then you go to the next frame and repeat the process in this way you will result in a top-view video feed. If there are speed limitations in your own implementation of warping you can use the OpenCV implementation instead. Your final visualisation should show live feed from all the cameras as well as the generate top-view video

Task 4: Visualise Object Detection on Orthographic Top-View

4.1 Repeat task 2.4

4.2 Object detector will give you a list of bounding boxes, note the mid-point of the base of bounding box and project it on to the top view/satellite view using the computed homography

4.3 Save all the top view locations in a separate file. You need to perform all the above mentioned steps for each view separately for both online mode as well as offline mode. Your final visualisation should show live feed from all the cameras with overland object detection results as well as the visualisation on the top-view

Task 5: Heatmap Visualization Top-View

5.1 **Static Heatmap:** Using all the recorded mid-point location on the top-view from all the camera generate a heat map. Heat map can be generated by placing a Gaussian with certain mean and sigma at each top-view location. All the Gaussian for all the locations for all the frames should be summed up and normalised by the number of time steps to generate a single heat map.

5.2 **Animated Heatmap:** In this task you are required to repeat task 5.1 but only for last *k* time steps i.e. from time step *t-k-1* to time step *t*. At time step *t* the heat map should exclude the data from time step *t-k* and instead include the data for time step *t*. This will result in updating the heat map for every time frame thus result in a animated heat map.

Task 6: Trip wire (Optional / Bonus)

6.1 Mark a polygon on either the camera view or the top view and generate an alert every time an activity is observed within the marked polygon. The alerts should be visualised on the live feed and should be save in the file as well. There should be a script/program that allows to visualise these alerts as well as short clips.

#	Description	Deadline on LMS	Submission Marks	Marks	Intermediate Evaluations Meetings
0:	Proposal	23 Oct	5%		
1:	Multi-camera Setup & Recording	02 Nov	5%		
2:	Object Detection (Person or Car)	09 Nov	5%	20%	1st Evaluation - Task 1 & 2
3:	Orthographics Top-View Generation	16 Nov	5%		
4:	Visualise Object Detection on Orthographic Top-View	23 Nov	5%	25%	2nd Evaluation - Task 3 & 4
5:	Heatmap Visualization Top-View	30 Nov	5%		
6:	Trip wire (Optional / Bonus)	07 Dec	5%	30%	Final Evaluation - Task 5 & 6
		Total	35%	75%	

Each submission on LMS is 5% weightage, evaluation meeting are weighted 20%, 25% and 30% respectively. In case you failed to complete the task by deadline on LMS your efforts will still be evaluated in the meetings. 2nd evaluation is for the progress on all the task till that date, similarly final evaluation is for the entire project and not just for heat map visualisation.