

COMP9517: Computer Vision

2022 Term 2

Group Project Specification

Maximum Marks Achievable: 40

The group project is **worth 40% of the total course marks**.

Project work is in Weeks 6-10 with a demo and report due in Week 10.
Refer to the separate marking criteria for detailed information on marking.
Submission instructions and a demo schedule will be released later.

Introduction

The goal of the group project is to work together with peers in a team of 4-5 students to solve a computer vision problem and present the solution in both oral and written form.

Each group can meet with their assigned tutor pair once per week in Weeks 6-9 during the usual consultation session on Fridays 2-3 PM to discuss progress and get feedback.

The group project is to be completed by each group separately. Do not copy ideas or any materials from other groups. If you use publicly available methods or software for some of the tasks, these must be attributed/referenced appropriately (failing to do so is plagiarism and will be penalised according to UNSW rules described in the Course Outline).

Description

An important and challenging computer vision task is object tracking in real-time videos or time-lapse image sequences [1-9]. Example applications include crowd surveillance, traffic monitoring, autonomous driving and flying, robotics, ocean and space exploration, precision surgery, and biology. In many applications, the large volume and complexity of such data make it impossible for humans to perform accurate, complete, efficient, and reproducible recognition and analysis of the relevant information in the data.

There are three fundamental steps in object tracking: object detection/segmentation in each frame of the video, object linking from frame to frame in order to obtain the trajectories, and object motion analysis from the trajectories. The difficulty in many applications is that objects may enter or leave the scene, touch/occlude each other, have similar appearance, and change appearance over time due to illumination changes, scale and shape changes, and deformations, making it hard to keep track of their unique identity. Therefore, object tracking is still a highly active research area in computer vision.

The goal of this group project is to develop and evaluate a method for tracking pedestrians and analysing their motion in real-world video recordings. Many traditional and/or machine or deep learning-based computer vision methods could be used for this. You are challenged to use the concepts taught in this course as well as other methods from literature [1-9] to create and implement your own tracking method and evaluate its performance on a public dataset from a recent international benchmarking study [10].

Tasks

The group project consists of three tasks described below, each of which needs to be completed as a group and will be evaluated for the whole group.

Public Dataset

The dataset to be used in the group project is from the Segmenting and Tracking Every Pixel (STEP) benchmark and consists of two training videos and two test videos. It is part of the long-standing Multiple Object Tracking (MOT) benchmark and provides annotations where every pixel has a semantic label and all pixels belonging to the most salient object class (pedestrian) have a unique tracking ID. The benchmark is part of the STEP-Workshop organised at the 2021 International Conference on Computer Vision (ICCV).

The dataset including the annotation labels and further information can be found here:

<https://motchallenge.net/data/STEP-ICCV21/>

The two training videos with corresponding annotations can be used to learn more about the data and (if you are using machine/deep learning) to train your method. For testing, you are required to demonstrate your method on the first test video. You are welcome to also demonstrate it on the second test video, but this is not required (it is a more difficult case).

Task 1: Track Pedestrians

Develop a Python program to track all pedestrians in the videos. Specifically, the program must perform the following subtasks:

- 1.1 Detect all pedestrians in all frames and calculate the bounding box for each of them. It is not necessary to perform pedestrian segmentation (though you are welcome to try). Notice this means the annotations (labels) of the training set provide more information (pixel-level) than needed for this project (object-level). To get training data for the detection task, you need to convert the pixel-label maps to bounding boxes.
- 1.2 Link the bounding boxes over time to obtain the trajectory for each pedestrian. This means identifying which detections in two successive frames of the video belong to the same pedestrian. Criteria for this can be based on distances between the boxes or features calculated from the pixel values within the boxes.
- 1.3 Draw the bounding box and corresponding trajectory for each pedestrian. That is, for each video frame, the program must show for each pedestrian in that frame its box at

that time point and its trajectory up to that time point. Use a unique colour per pedestrian to draw the box and trajectory. The trajectory can be drawn for example as a piecewise linear curve connecting the centre positions of the corresponding boxes, from the time when the pedestrian first appeared up to the current time point.

Task 2: Count Pedestrians

Extend the program so that it can count the number of pedestrians over time. Specifically, the program must perform the following subtasks:

- 2.1 Report the total count of all unique pedestrians detected since the start of the video.
- 2.2 Report the total count of pedestrians present in the current video frame.
- 2.3 Allow the user to manually draw a rectangular region within the video window.
- 2.4 Report the total count of pedestrians who are currently within that region.

The counts can be reported by printing them to the terminal or (better) directly on the video frame (for example in one of the corners of the window).

Task 3: Analyse Pedestrians

Further extend the program so that it can analyse the behaviour of pedestrians over time. Specifically, the program must perform the following subtasks:

- 3.1 Report how many pedestrians walk in groups and how many walk alone. Define a criterion to determine this from the bounding boxes.
- 3.2 Show occurrences of group formation and group destruction. A group formation event is when two or more pedestrians meet (get close) and stay together for more than one frame. A group destruction event is when at least one member of a group leaves.
- 3.3 Show occurrences of pedestrians entering or leaving the scene. For this subtask and the previous, use your creativity in automatically highlighting (drawing the observer's visual attention to) these events in the video.

Deliverables

The deliverables of the group project are 1) a group video demo and 2) a group report. Both are due in Week 10. More detailed information on the two deliverables:

Video Demo: Each group will prepare a video presentation of at most 10 minutes showing their work. The presentation must start with an introduction of the problem and then explain the used methods, show the obtained results, and discuss these results as well as ideas for future improvements. This part of the presentation should be in the form of a short PowerPoint slideshow. Following this part, the presentation should include a demonstration of the methods/software in action. Of course, some methods may take a long time to compute, so you may record a live demo and then edit it to stay within time.

The entire presentation must be in the form of a video (720p or 1080p mp4 format) of at

most 10 minutes (anything beyond that will be cut off). All group members must present (points may be deducted if this is not the case), but it is up to you to decide who presents which part (introduction, methods, results, discussion, demonstration). In order for us to verify that all group members are indeed presenting, each student presenting their part must be visible in a corner of the presentation (live recording, not a static head shot), and when they start presenting, they must mention their name.

Overlaying a webcam recording can be easily done using either the video recording functionality of PowerPoint itself (see for example [this tutorial](#)) or using other recording software such as OBS Studio, Camtasia, Adobe Premiere, and many others. It is up to you (depending on your preference and experience) which software to use, as long as the final video satisfies the requirements mentioned above.

During the scheduled lecture/consultation hours in Week 10, that is Tuesday 2 August 2022 9-11 AM and Friday 5 August 2022 1-3 PM, the video demos will be shown to the tutors and lecturers, who will mark them and will ask questions about them to the group members. Other students may tune in and ask questions as well. Therefore, all members of each group must be present when their video is shown. A roster will be made and released closer to Week 10, showing when each group is scheduled to present.

Report & Code: Each group will also submit a report (max. 10 pages, [2-column IEEE format](#)) along with the source codes, before 5 August 2022 18:00:00.

The report must be submitted as a PDF file and include:

1. Introduction: Discuss your understanding of the task specification and dataset.
2. Literature Review: Review relevant techniques in literature, along with any necessary background to understand the methods you selected.
3. Methods: Justify and explain the selection of the methods you implemented, using relevant references and theories where necessary.
4. Experimental Results: Explain the experimental setup you used to evaluate the performance of the developed methods and the results you obtained.
5. Discussion: Provide a discussion of the results and method performance, in particular reasons for any failures of the method (if applicable).
6. Conclusion: Summarise what worked / did not work and recommend future work.
7. References: List the literature references and other resources used in your work. All external sources (including websites) used in the project must be referenced.

The complete source code of the developed software must be submitted as a ZIP file and, together with the report, will be assessed by the markers. Therefore, the submission must include all necessary modules/information to easily run the code. Software that is hard to run or does not produce the demonstrated results will result in deduction of points.

Plagiarism detection software will be used to compare all submissions pairwise (including submissions for similar assignments in previous years, if applicable) for both the report and

the source code. See the Course Outline for the UNSW Plagiarism Policy.

As a group, you are free in how you divide the work among the group members, but all group members are supposed to contribute approximately equally to the project in terms of workload. An online survey will be held at the end of term allowing students to anonymously evaluate their group members' relative contributions to the project. The results will be reported only to the LIC and the Course Administrators, who at their discretion may moderate the final project mark for individual students if there is sufficient evidence that they contributed substantially less than the other group members.

References

The following papers provide much useful information about object tracking in computer vision. If the papers are not directly available (open access) by clicking the links, they should be available online via the UNSW Library.

- [1] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah. Visual tracking: an experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(7):1442-1468, July 2014. <https://doi.org/10.1109/TPAMI.2013.230>
- [2] P. Li, D. Wang, L. Wang, H. Lu. Deep visual tracking: review and experimental comparison. *Pattern Recognition* 76:323-338, April 2018. <https://doi.org/10.1016/j.patcog.2017.11.007>
- [3] M. Fiaz, A. Mahmood, S. Javed, S. Jung. Handcrafted and deep trackers: recent visual object tracking approaches and trends. *ACM Computing Surveys* 52(2):43, April 2019. <https://doi.org/10.1145/3309665>
- [4] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, F. Herrera. Deep learning in video multi-object tracking: a survey. *Neurocomputing* 381:61-88, March 2020. <https://doi.org/10.1016/j.neucom.2019.11.023>
- [5] M. Y. Abbass MY, K.-C. Kwon, N. Kim, S. A. Abdelwahab, F. E. A. El-Samie, A. A. M. Khalaf. A survey on online learning for visual tracking. *The Visual Computer* 37(5):993-1014, May 2021. <https://doi.org/10.1007/s00371-020-01848-y>
- [6] Y. Zhang, T. Wang, K. Liu, B. Zhang, L. Chen. Recent advances of single-object tracking methods: a brief survey. *Neurocomputing* 455:1-11, September 2021. <https://doi.org/10.1016/j.neucom.2021.05.011>
- [7] E. Meijering, O. Dzyubachyk, I. Smal, W. A. van Cappellen. Tracking in cell and developmental biology. *Seminars in Cell and Developmental Biology* 20(8):894-902, October 2009. <https://doi.org/10.1016/j.semcdb.2009.07.004>
- [8] D. Chaudhary, S. Kumar, V. S. Dhaka. Video based human crowd analysis using machine learning: a survey. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 10(2):113-131, October 2021. <https://doi.org/10.1080/21681163.2021.1986859>
- [9] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, S. Kasaei. Deep learning for visual tracking: a comprehensive survey. *IEEE Transactions on Intelligent Transportation Systems* 23(5):3943-3968, May 2022. <https://doi.org/10.1109/TITS.2020.3046478>
- [10] M. Weber, J. Xie, M. Collins, Y. Zhu, P. Voigtlaender, H. Adam, B. Green, A. Geiger, B. Leibe, D. Cremers, A. Ošep, L. Leal-Taixé, L.-C. Chen. STEP: segmenting and tracking every pixel. *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS)*, December 2021.

<https://doi.org/10.48550/arXiv.2102.11859>

Copyright: UNSW CSE COMP9517 Team

Released: 28 June 2022