CS 484/555

Fall 2024

Homework Assignment 3

Due: 21 December 23:59

1 Objective

This assignment focuses on implementing two key algorithms for image alignment and feature tracking. These algorithms find extensive applications in video stabilization, motion estimation, and object tracking in computer vision. By completing this assignment, you will gain practical experience in implementing, testing, and analyzing these techniques.

The provided dataset, detailed in Section 2, will be used for tracking a rectangular template image (in Sec 3) and corner points (in Sec 4) throughout a video sequence. Your task is to implement the following algorithms by yourself:

1. **Lucas-Kanade Forward Additive Alignment (Translation Only)**
2. **Kanade-Lucas-Tomasi (KLT) Tracker (Translation Only**)

and write a report regarding your implementation. What is expected from this report is detailed in section 5 in detail.

2 Dataset

You are provided with three video sequences, along with a rectangular template image extracted from the first frame of each video. The video frames and corresponding templates can be accessed in the assignment folder. The dataset frames can be seen in Fig. 2, 3, 4 and the rectangle templates can be seen in Fig 5. You can reach the video frames and the rectangle templates from the assignment folder. In addition, the position of the template image extracted from the first frame is also provided as part of the assignment in terms of:

- x coordinate of the upper left corner
- y coordinate of the upper left corner
- width
- height

in the file **crop_metadata.txt**. An example visualization for this format is provided in Fig 1.

3 Lucas-Kanade Forward Additive Alignment [30 points]

In this section, you will implement a simple Lucas & Kanade tracker with a single template. In the scenario of two-dimensional tracking with a pure translation warp function,

$$\mathbf{W}(\mathbf{x};\mathbf{p}) = \mathbf{x} + \mathbf{p}. \tag{3.1}$$

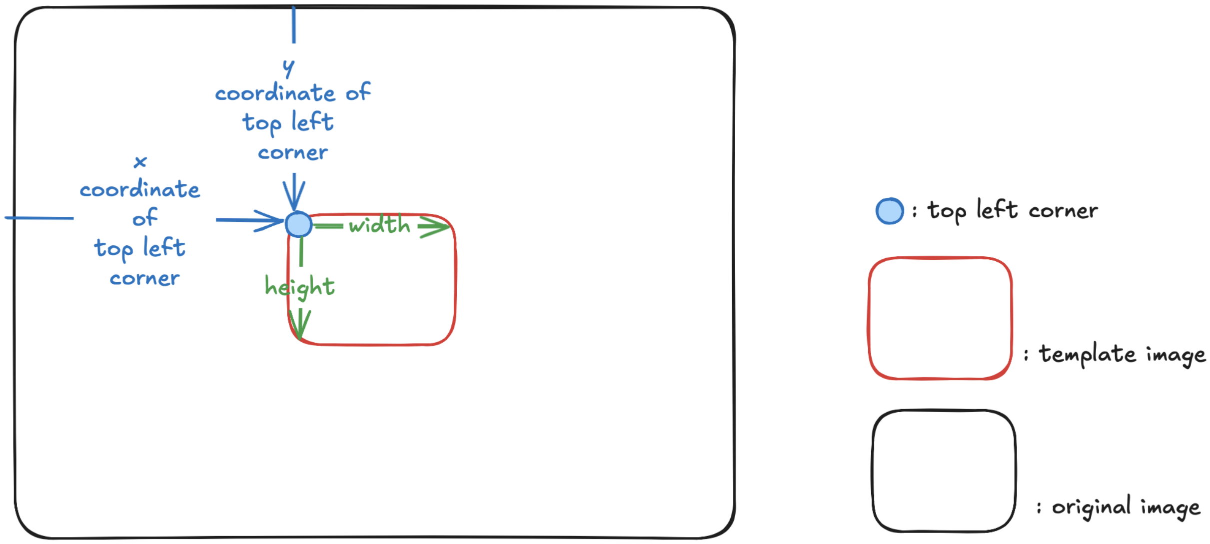

Figure 1: Diagram of the position data format of the template image

Figure 2: First Image set of assignment.

Figure 3: Second Image set of assignment.

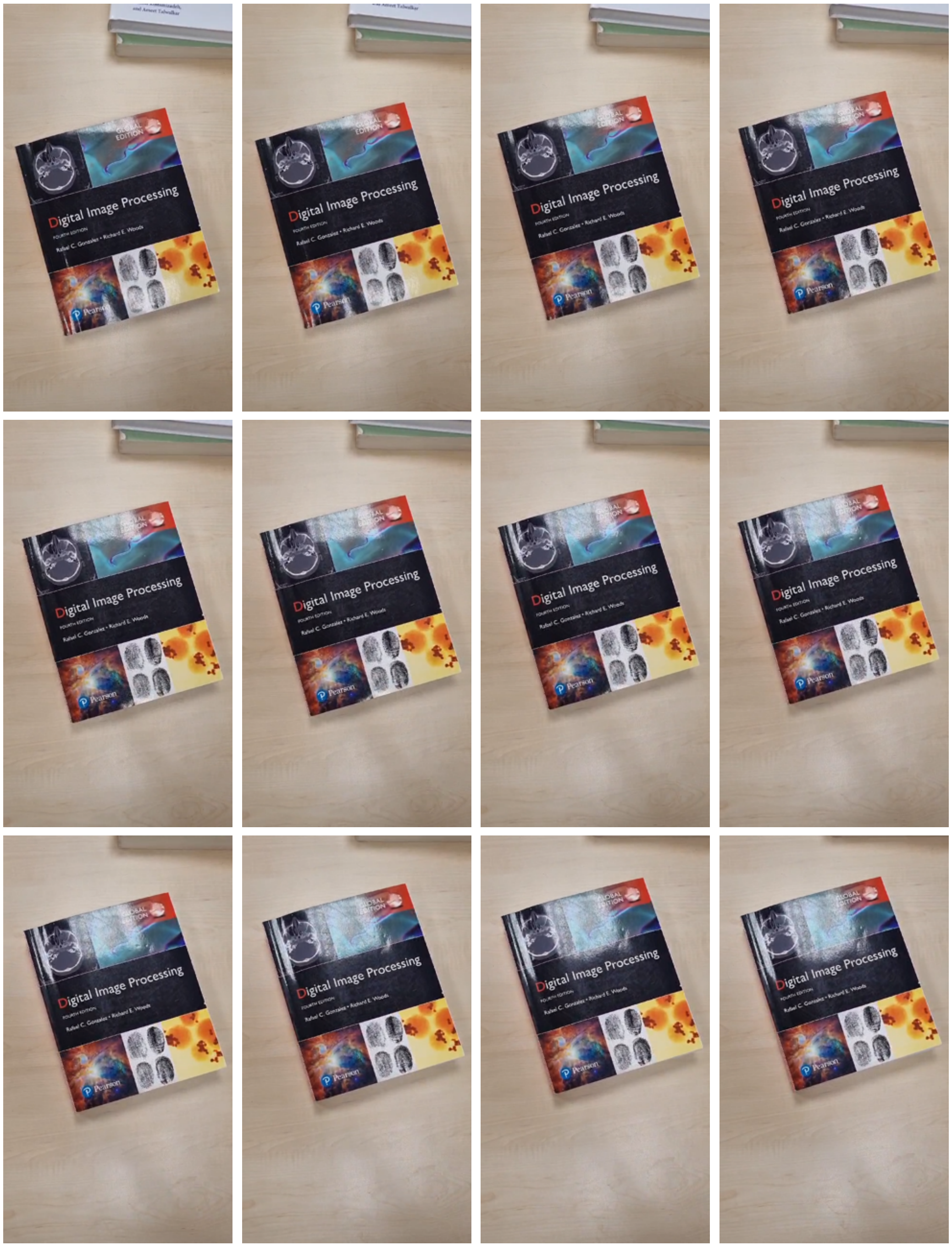

Figure 4: Third Image set of assignment.

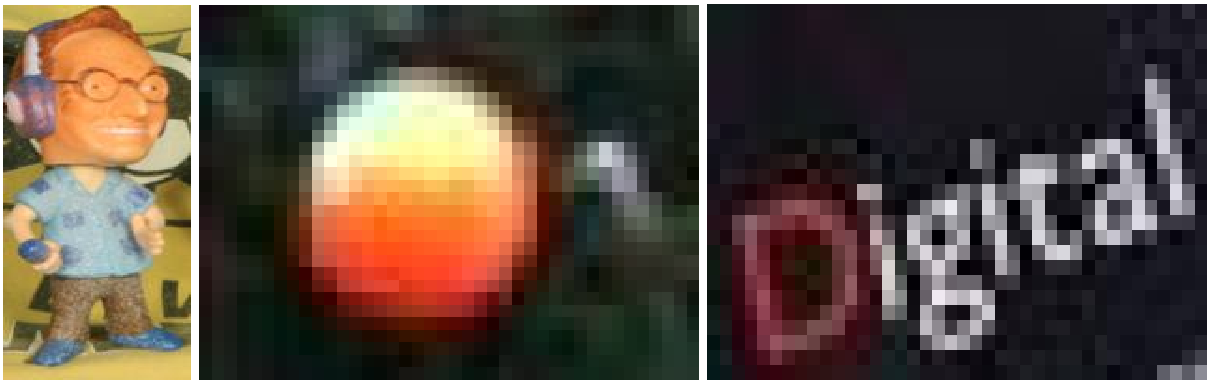

Figure 5: Rectangular template images to be used for each video sequence in the assignment

The problem can be described as follows: Starting with a rectangular neighborhood of pixels $\mathcal{N} \in \{\mathbf{x}_d\}_{d=1}^{D}$ in frame I_t, the Lucas-Kanade tracker aims to move it by an offset $\mathbf{p} = [p_x, p_y]^T$ to obtain another rectangle on frame I_{t+1}, so that the squared pixel difference in the two rectangles is minimized:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \sum_{\mathbf{x}\in\mathcal{N}} \|I_{t+1}(\mathbf{x}+\mathbf{p}) - I_t(\mathbf{x})\|_2^2 \tag{3.2}$$

$$= \left\| \begin{bmatrix} I_{t+1}(\mathbf{x}_1+\mathbf{p}) \\ \vdots \\ I_{t+1}(\mathbf{x}_D+\mathbf{p}) \end{bmatrix} - \begin{bmatrix} I_t(\mathbf{x}_1) \\ \vdots \\ I_t(\mathbf{x}_D) \end{bmatrix} \right\|_2^2. \tag{3.3}$$

Starting with an initial guess of $\mathbf{p}$ (for example, $\mathbf{p} = [0,0]^\top$), we can iteratively calculate the optimal $\mathbf{p}^*$. In each iteration, the objective function is locally linearized by the first-order Taylor expansion.

$$I_{t+1}(\mathbf{x}' + \Delta\mathbf{p}) \approx I_{t+1}(\mathbf{x}') + \nabla I_{t+1}(\mathbf{x}') \frac{\partial \mathbf{W}(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}} \Delta\mathbf{p}, \tag{3.4}$$

where $\Delta\mathbf{p} = [\Delta p_x, \Delta p_y]^\top$, is the template offset. In addition, $\mathbf{x}' = \mathbf{W}(\mathbf{x};\mathbf{p}) = \mathbf{x}+\mathbf{p}$ and $\nabla I(\mathbf{x}')$ is a vector of the x- and y-image gradients at the pixel coordinate $\mathbf{x}'$. From your lecture notes uploaded on Moodle, one can incorporate these linearized approximations into a vectorized form such that

$$\arg\min_{\Delta\mathbf{p}} \|A\Delta\mathbf{p} - \mathbf{b}\|_2^2, \tag{3.5}$$

to compute $\Delta\mathbf{p}$ and update $\mathbf{p}$ as $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$ at each iteration. Using this information, and the coordinates of the template object in the first frame, find the coordinates of the objects in all frames of the videos present in the given rectangular region in the dataset. You are required to experiment with different values of ϵ (convergence threshold) which can be used to stop the iterative updates when the error is below this value and the maximum number of iterations. The goal is to find the best-fit values that balance accuracy and efficiency for the dataset.

Hint: While no specific values for ϵ and the number of iterations are provided, you are encouraged to begin with reasonable estimates and iteratively adjust them based on the observed results. Ensure that your report includes a discussion of the chosen values and their impact on the performance of the algorithms.

4 Kanade-Lucas-Tomasi (KLT) Tracker [40 points]

The **Kanade-Lucas-Tomasi (KLT) tracker** is an efficient method for tracking small templates or corner points between consecutive frames, using a simple translational motion model. In this part of the assignment, you will implement the KLT tracker with the following steps:

1. **Corner Detection:**
   - Identify corner points in the first frame of the video sequence using a corner detection algorithm.
   - The corner detection algorithm (Harris Corner Detector) must be implemented by you. However, you may use external libraries for linear algebra operations such as eigenvalue calculation, determinant, and trace computations.
   - Select only those corner points that lie within the bounding box of the region of interest provided as part of the assignment.
2. **Corner Tracking:**
   - For each selected corner point, calculate its displacement to the next frame using the Lucas-Kanade method.
   - Update the positions of the corners based on the calculated displacements.
   - Using the tracked corner points, estimate the position of the bounding box in the next frame.
   - You may periodically add new corner points (e.g., every few frames) to maintain a sufficient number of tracked points. Make sure that you have at least 4-5 corner points for each tracking operation.
3. **Iterative Updates and Parameter Tuning:**
   - Use a translational warp function to iteratively update the corner positions until convergence.
   - Experiment with different values for ϵ (the convergence threshold) and the maximum number of iterations to optimize the tracking performance.
   - The threshold ϵ determines when the iterative updates should stop based on the error reduction. Choose values that balance accuracy and computational efficiency.

Hint:

- Reasonable starting estimates for ϵ and iteration limits can guide your experiments.
- Iteratively refine these parameters based on the observed results.
- Document the chosen values and their effects on accuracy and efficiency in your report, providing a clear analysis of how they influenced the performance of the KLT tracker.

5 Report [30 points]

In addition to the experimental result and the code, we expect you to provide a report in your submission. In your report, make sure to include the sections discussed below.

Introduction: Provide an overview of the assignment objectives and describe the two algorithms implemented: Lucas-Kanade Forward Additive Alignment, and Kanade-Lucas-Tomasi (KLT) Tracker, along with their core differences. Mention the significance of experimenting with different ϵ (convergence threshold) and iteration values to achieve optimal performance.

Methodology

- **Lucas-Kanade Forward Additive Alignment:** Describe the mathematical formulation of the algorithm, including key equations. Provide a step-by-step explanation of your implementation, challenges faced, and how they were addressed. Include your approach to experimenting with different ϵ and iteration values.
- **Kanade-Lucas-Tomasi (KLT) Tracker:** Describe the mathematical formulation of the algorithm, including key equations. Provide a step-by-step explanation of your implementation, challenges faced, and how they were addressed. Highlight how you experimented with different ϵ and iteration values to optimize the tracking process.

Experimental Results

- **Visualization of Tracking Results:** Include images of the tracked bounding boxes or corner points overlaid on video frames for each dataset and algorithm. Provide the number of iterations required for convergence for each algorithm under different ϵ selections and discuss the observed trends.
- **Comparative Analysis:** Discuss the relative performance of the algorithms in terms of accuracy and efficiency. Highlight how varying ϵ and iteration counts impacted the performance. Identify cases where specific algorithms performed better or worse and explain why.

  NOTE: As we have not provided you with the original coordinates of the template image for later frames, we expect you to assess accuracy intuitively using visual inspection.

Discussion

- Discuss the strengths and limitations of each algorithm, particularly in the context of different ϵ and iteration values.
- Suggest potential improvements or alternative approaches to achieve better performance.

Conclusion: Summarize your findings, including the impact of different ϵ and iteration values on the algorithms. Reflect on the challenges faced and the key takeaways from the assignment.

References: Provide a list of all resources used, including lecture notes, textbooks, and any other references.

6 Notes

- You should submit your solutions as a single archive file that contains your code and report to Moodle.
- You can use 3rd party libraries for reading images and matrix operations. However, all the core operations related to the tracking algorithm should be implemented by **YOURSELF** without any LLM guidance or 3rd party library usage.
- For any questions, please contact yigit.ekin@bilkent.edu.tr by email.