

Implementation of the Kanade-Lucas-Tomasi Feature Tracker with Python

Weston Scott
University of Arizona
scottwj@arizona.edu

Abstract—Feature tracking is a fundamental aspect of computer vision, enabling the analysis of dynamic scenes. The Kanade-Lucas-Tomasi (KLT) feature tracker is a well-established method for tracking features across frames in video sequences. This project aims to implement the KLT tracker using Python, evaluate its performance against other tracking algorithms, and analyze its efficiency and accuracy in various scenarios in the aerial object tracking domain.

I. INTRODUCTION

Feature tracking is essential in various applications, including object recognition, motion analysis, and augmented reality. The Kanade-Lucas-Tomasi (KLT) feature tracker, based on the principles of optical flow and feature detection, provides a robust method for tracking points in video sequences. This project will involve implementing the KLT tracker in Python and comparing its performance to alternative tracking methods.

II. PROJECT DESCRIPTION

The proposed project will include the following steps:

- **Understanding the KLT Algorithm:** Review the principles of the KLT tracker, including its reliance on optical flow and feature detection techniques.
- **Implementation:** Develop a Python implementation of the KLT tracker from scratch using libraries such as OpenCV and NumPy, following these key steps:
 - 1) Convert input video frames to grayscale.
 - 2) Compute gradients of the image using Sobel operators (convolution kernels)
 - 3) Calculates scores for each pixel, helping identify strong corners.
 - 4) Detect key features in the first frame (and subsequent frames) using the Shi-Tomasi corner detector, which scores pixels.
 - 5) Select corners based on scores.
 - 6) Apply the Lucas-Kanade optical flow method to track features in subsequent frames which calculates the displacement of corners between frames.
 - 7) Visualize tracked features on the video frames.
- **Variations:** Implementation varying types/ sizes of convolution kernels for the sobel operators will be explored, in addition to image sharpening/blurring techniques.
- **Comparative Analysis:** Implement an alternative tracking method for performance comparison, such as the mean-shift algorithm, as well as built-in python implementations of the KLT algorithm to compare with.

III. REQUIRED RESOURCES

The project will require:

- **Software:** Python 3.X (probably 3.12 or above), OpenCV, NumPy, Matplotlib, Scipy.
- **Computing Resources:** Linux-based operating system with sufficient storage and memory to handle video processing.
- **Test Data:** Publicly available video datasets of drones or other aircrafts following boats or other aerial vehicles (<https://www.pexels.com/search/videos/>). Processing the data will be required to ensure truth data can be measured against.

IV. IMPLEMENTATION PLAN

The plan is to use existing Python libraries (OpenCV) to load videos and convert them to grayscale, as well as display them after algorithm processing. The rest of the steps outlined in the project description are planned to be implemented from scratch using Numpy. Though this may be computationally inefficient, it will demonstrate the step-by-step process of how the KLT algorithm works.

V. TESTING AND EVALUATION

To validate the implementation, the algorithm developed from scratch will be benchmarked against existing library implementations of the KLT algorithm, and other algorithms such as the mean-shift algorithm.

VI. CONCLUSION

This project will enhance understanding of feature tracking techniques while providing hands-on experience in implementing and evaluating a well-established algorithm in computer vision. By comparing the KLT tracker with alternative methods, the project will highlight the practical implications of feature tracking in real-world applications.

REFERENCES

- [1] L. Tan, *Digital Signal Processing*, INAOEP, 2013. [Online]. Available: https://www-elec.inaoep.mx/~jmram/Digital_Signal_Processing__LI_TAN.pdf.
- [2] "Kanade-Lucas-Tomasi feature tracker," *Wikipedia*, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Kanade%E2%80%93Lucas%E2%80%93Tomasi_feature_tracker.
- [3] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, 3rd ed., Pearson, 2010.
- [4] C. Kanade, T. Lucas, and B. Tomasi, "An improved optical flow algorithm," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1987, pp. 12-21. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6716443>.