The KLT Tracker

Lab assignment 2

1 Introduction

During this assignment, you will apply the OpenCV version of the widely used Kanade-Lucas-Tomasi tracker (KLT tracker) [2, 3] within a ROS framework. From a streaming video, a selected region-of-interest (ROI) is to be tracked, using a number of feature points and the KLT tracker. For the lab assignment, the ROI can be selected manually, but for the challenge it will be required to select it based on the markers (supply boxes) or a person detector (victims).

2 Preliminaries

You already know from previous parts of the course how to set up a ROS system and how to build nodes, define topics etc. All this knowledge will be required for the present assignment, as well as access to a web-cam functioning in ROS.

2.1 Useful packages and functions

The ROS package video_stream_opencv `http://wiki.ros.org/video_stream_opencv` should be considered to produce a topic with a video stream.

From OpenCV, the following functions will be useful:

- `goodFeaturesToTrack` for adaptively sampling points in the ROI [3]
- `calcOpticalFlowPyrLK` for KLT tracking [2]
- `HOGDescriptor` for detecting people [1] (optional for the lab)

2.2 Preparation

Read the documentation of the packages and functions above. Think about the ROS graph for the addressed problem to track *one* object and the control structure for the node(s) to be developed.[1]

Task 1: Draw a flow chart containing at least one (abstract) *ROI detector*, one point *sampler*, and point *trackers*. What are the conditions to activate the respective modules? Note that tracked points may get lost.

[1] From now on, this document will only refer to *node* in singular. If you choose to implement your algorithm with more than one node, you may replace it with *nodes* whenever suitable.

3 Implementation, test, and system integration

Once you have designed an algorithm to address the tracking problem, the necessary node needs to be implemented, tested, and integrated into a minimal system to solve the task.

3.1 Node implementation

Implement your node. Note that parameters should not be set inside your code, but as command line parameters or via the NodeHandle (e.g. for the use in a launch file). Your node should continuously report the position of the target.

Task 2: What parameters are required for your node? What topics does your node subscribe to? What topics does your node publish?

Your node may select feature points by regular sampling on a grid or by `goodFeaturesToTrack`.

Task 3: What are the advantages and drawbacks of these methods?

3.2 Testing

Using your node directly with a web-cam might lead to failures that are difficult to reproduce. For testing purposes, it might be better to generate a synthetic test sequence that your node subscribes to. Note that noise in the image data might lead spurious estimated motions even if the motion is zero.

Task 4: What is a suitable test sequence? What should be varied in the test sequence? How can the spurious motion caused by noise be avoided?

3.3 System integration

Build a minimal system around your node to produce a small demo: In the first frame from the image stream, a ROI is marked manually. As soon as it is marked, the algorithm will continue to track that ROI. You may assume that the ROI has not moved significantly between the initial frame and the first frame to be tracked.

Task 5: What is avoided by the assumption? What do you observe if the object rotates? What do you observe if the lighting conditions change?

Optional: replace the manual detector with a person detector and repeat the experiments with a person in the camera stream.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005.

[2] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of International Joint Conference on Artificial Intelligence*, 1981.

[3] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.