

1 TLD Implementattion

TLD implementation is based on three main modules such as: Tracker, Detector and Learner (Integrator). Tracker and Detector modules are completely independent from each other. However, Integrator combines two separate results from Tracker and Detector, validates and sends feedback to detector in order to rule out false positives and false negatives.

1.1 Tracker

Tracker takes two consequent frames and a bounding box contains the target object within the first image. Trackers main task is that to estimate the location of the bounding box in the current frame or to decide whether the target object is still visible. Basically, tracker generates points in the given bounding box, tracks each point from frame to frame, filter the reliable ones and estimates the overall displacement with reliable points.

Point Generation In order to generate points in the given box, grid point approach is used. With a margin to each border of box, put a grid with 10 grid points in both direction: width and height.

Point Tracking TLD uses pyramidal lucas-kanade tracker, which is already implemented in OpenCV, to track each point from frame to frame; but instead of tracking one way, we track the points, previous operation has end up, backwards. This forward-backward tracking gives us a good metric for evaluation of reliability of points. **I can explain flags and parameters in calcOpticalFlowPyrLK?**

Filtering Reliable Points At this point, we have point to point correspondences with forward backward error. However, we still do not know which points are really tracked successfully. This is why, we apply normalized cross correlation between points to decide that still coordinates point to same location of object or shifted away. Besides ncc error, we already have forward-backward error. With these error metrics, we apply median filtering to points and eliminate outliers.

Relocation of Box At the end of the tracking process, we have points and relatively robust displacements for each of them. With these displacements, we, again, get the median value of displacements in two direction and to the coordinates of box respectively.

1.2 Detector

Detector also takes the current frame and returns couple of bounding boxes which are probably includes the instances of the object. Since cascaded model is used in design of the detector, there are multi layers that detector consist of.

- What is the overall flow to generate possible boxes from given frame?
- What are the layers/steps in cascaded detector?
- What are responsibilities and aim of the each step?

1.3 Learner

Learners job is to reason out the tracking result and detection result.

- How to validate when Tracker tracks given box?
- How to decide whether detector or tracker is valid?
- All possible cases, etc?