Fusion of Tracking Techniques to Enhance Adaptive Real-Time Tracking of Arbitrary Objects

Peter Poschmann, Patrik Huber, Matthias Rätsch, Joseph Kittler, Hans-Joachim Böhme

HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT DRESDEN
UNIVERSITY OF APPLIED SCIENCES

Problem

Our goal is to determine the location of a target in each frame of an image sequence, to detect its disappearance, and to be able to re-detect it after an occlusion. Without prior knowledge, the tracker has to adapt to the target, background and recording conditions of the video. Each update introduces some error, so the tracker might drift away from the target over time.

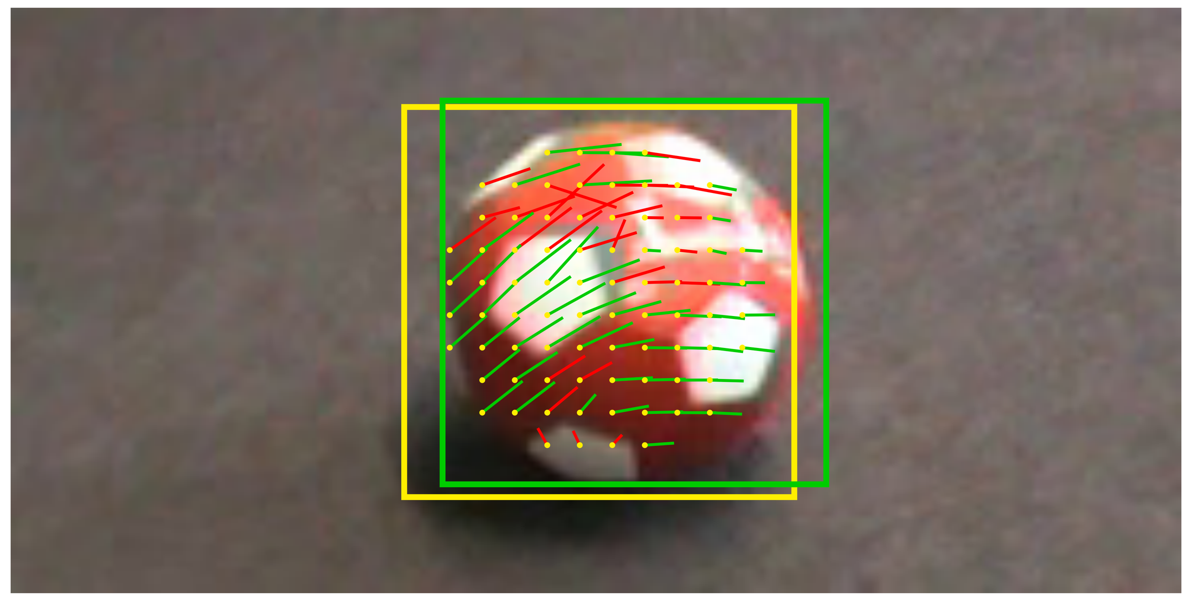
Method

We present an adaptive tracker that uses a particle filter to track position and size of the target. To increase its robustness and accuracy, we tested three enhancements: An optical-flow-based motion estimation, a learning strategy for preventing bad updates while staying adaptive, and a sliding window detector for fast re-detection and finding the best negative training examples.

Source Code

The source code, configurations and detailled results of our tracking algorithm are available at `http://adaptivetracking.github.io/`

Motion Model

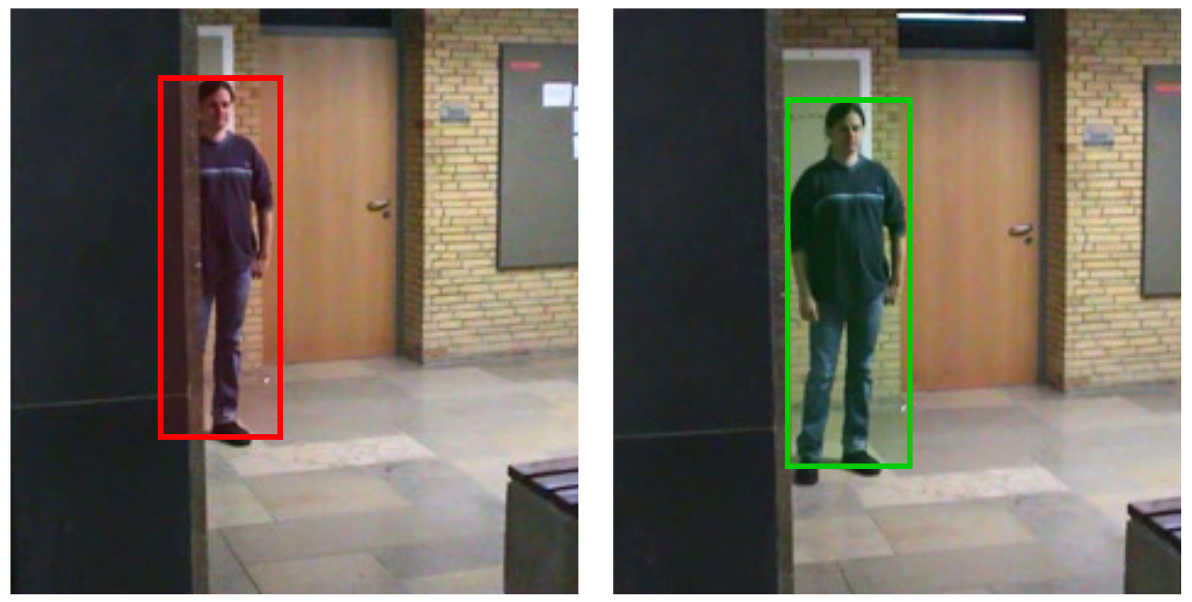


- Baseline: Constant velocity model
- Enhancement: Predict new target position using optical flow of a grid of points
- Remove bad points (red lines in figure above) [1]
- Optical flow improves the performance in most sequences, but fails in case of out-of-plane rotations or (partial) occlusions

Measurement Model

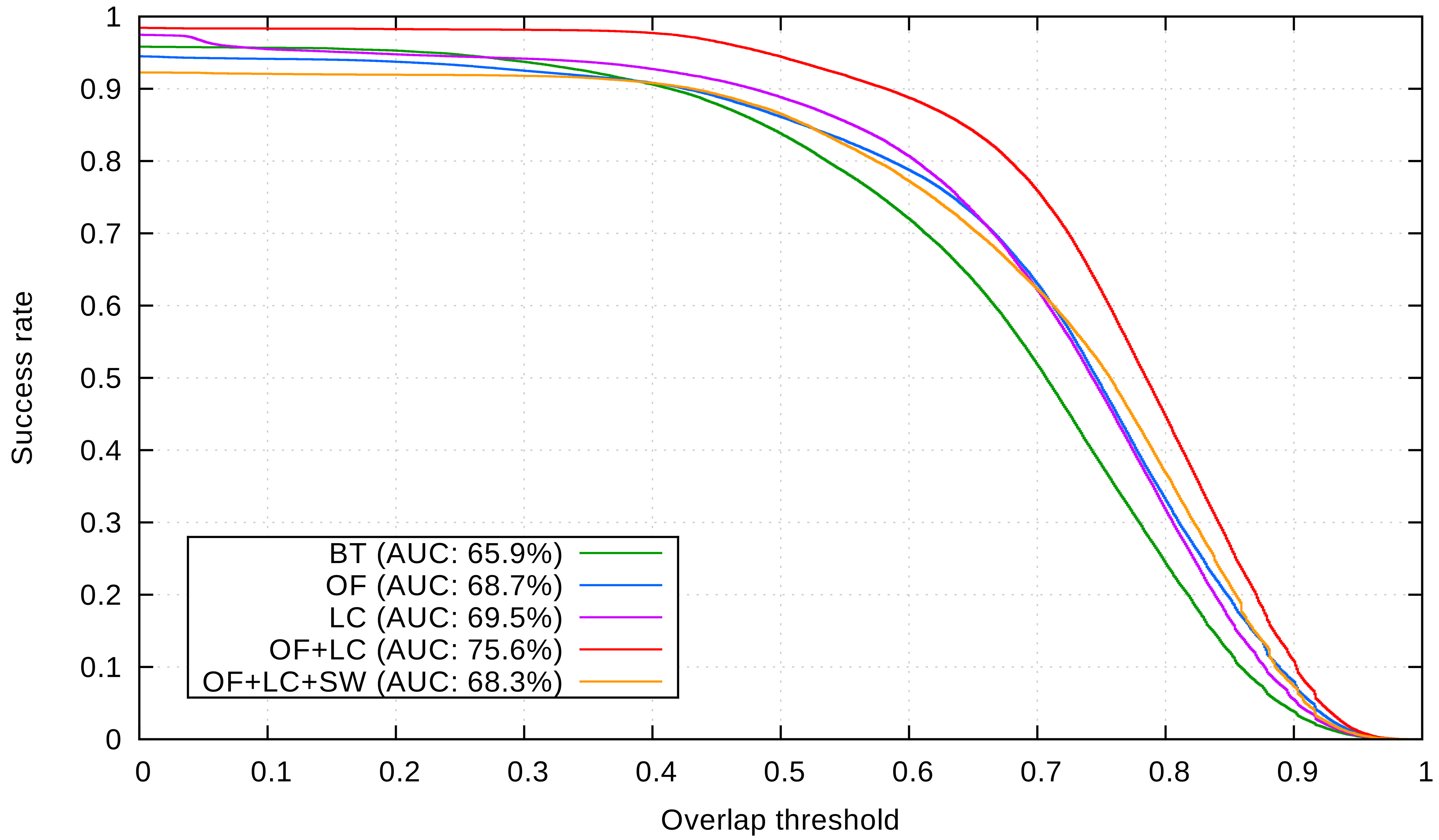
- Compute particle weights from linear SVM score of extended HOG features [2]
- Baseline: Features computed for each particle
- Enhancement: Sliding window detector on feature pyramid for fast re-detection after occlusions, failure detection, and to find the best negative training examples in each frame
- Particles get SVM score from detector response
- Sliding-window-based measurement model is less accurate (resolution is at HOG cell instead of pixel level), yellow particle in figure above will be weighted as if it was at the green position
- Increases run-time efficiency, as convolution on all feature pyramid layers is faster than computing features and score per particle

Update Strategy



- Baseline: Update SVM whenever the estimated target position is considered positive
- Enhancement: Introduce learning condition, update SVM only when the score is above some predefined threshold
- Learning condition improves performance by reducing bad updates (e.g. partial occlusions, see left figure), but in some cases prevents all updates

Results



- Evaluated using BoBoT dataset [3, 4] consisting of 13 image sequences (320x240 pixels at 25 fps)
- Tracking of pedestrians and arbitrary objects
- Features conditions like lighting changes, partial and full occlusions, out-of-plane rotations, changing background, and distracting objects
- Diagram shows fraction of correctly tracked frames given a chosen overlap threshold
- Overlap is the fraction of intersection to union of ground truth and estimated bounding box

- *BT:* Base tracker
- *OF:* Optical-flow-based motion model
- *LC:* Learning condition
- *SW:* Sliding-window-based measurement model
- Both optical-flow-based motion model and learning condition increase performance, especially when combined
- Adding the sliding-window-based measurement model deteriorates performance, but increases speed (∼60 fps compared to ∼25 fps)

Further Work

- Choice of learning threshold is crucial
- Too low: Bad updates, drifts away from target
- Too high: No updates, loses target after appearance changes
- Ideal threshold depends on conditions, a single threshold does not work equally well on all image sequences
- We will explore the possibility of having an adaptive threshold or find other ways to prevent bad updates and allow correct ones

References

[1] Z. Kalal, K. Mikolajczyk, J. Matas. Forward-backward error: Automatic detection of tracking failures. In *ICPR* 2010, p. 2756–2759.

[2] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan. Object detection with discriminatively trained part-based models. In *IEEE TPAMI* 2010;**32**(9):1627–1645.

[3] D.A. Klein, D. Schulz, S. Frintrop, A.B. Cremers. Adaptive real-time video-tracking for arbitrary objects. In *IROS* 2010, p. 772–777.

[4] D.A. Klein, A.B. Cremers. Boosting scalable gradient features for adaptive real-time tracking. In *ICRA* 2011, p. 4411–4416.