– Supplementary Material –
Appearances can be deceiving: Learning visual tracking from few trajectory annotations

Santiago Manen[1], Junseok Kwon[1], Matthieu Guillaumin[1], and Luc Van Gool[1,2]

[1] Computer Vision Laboratory, ETH Zurich
[2] ESAT - PSI / IBBT, K.U. Leuven
{smanenfr, kwonj, guillaumin, vangool}@vision.ee.ethz.ch

In this supplementary material, we present additional details for some aspects of our work. We detail in Sec. 1 the properties of the optimization problem that we solve for learning how to track, and which has lead to the algorithm proposed in the paper. Then we show in Sec. 2 all the plots on all the datasets.

1 Deceptiveness Learning Properties

In this section, we discuss the properties of the optimization problem in eq. (5) in the main paper (Sect. 4.1):

$$\begin{aligned} \underset{\widetilde{\boldsymbol{\rho}}}{\text{minimize}} \quad & \sum_{k=1}^{T} \widetilde{\rho}_k \\ \text{subject to} \quad & \forall k \in [1, T], \quad 0 \leq \widetilde{\rho}_k \leq 1, \\ & \text{trackingError}(\widetilde{\boldsymbol{\rho}}) \leq \theta, \end{aligned}$$

At first, this optimization might look very general, however a closer look reveals some interesting properties:

- For all $\theta \geq 0$, there is always a solution to eq. (5): $\widetilde{\boldsymbol{\rho}} = 1$. That is, the tracker completely relies on the motion model, transferring exactly the only trajectory available which is the ground-truth trajectory of the object being tracked. That is: trackingError$=0$.
- trackingError depends on the full trajectory, so in principle we need the full track to compute it. However, at each frame k we can compute a lower bound on the final trackingError by looking at the error up to frame k:

$$\text{trackingError}_k = \sum_{i=1}^{k} \xi_i \,/T, \tag{1}$$

  where ξ_i is the center location distance between the window at frame i and its corresponding ground truth. Since trackingError_k is monotonically increasing, we can stop tracking as soon as trackingError_k reaches θ (we say that the tracking has failed at step k).

- Failure at step k (and more generally, trackingError_k) can only be recovered (resp. decreased) by increasing $\widetilde{\boldsymbol{\rho}}$ at a frame $k' < k$. That is, there is an earlier deceptive region not yet accounted for.
- When $\widetilde{\boldsymbol{\rho}}$ is modified at frame k, the track after frame k is altered. Any previous knowledge about the value of $\widetilde{\boldsymbol{\rho}}$ after frame k should be discarded.

These properties have lead us to the algorithm proposed in Sec. 4.1 of the main paper (paragraph *Optimization process*).

2 Complementary Plots

In the next page, we show plots that complement to the ones shown in the main paper (Sec. 6). Fig. 1 refers to the annotation selection approach, Fig. 2 to the conditioning on an event model and Fig. 3 on the learning of deceptiveness. They show the performance of our method compared to competitors or baselines for both datasets and both metrics, whereas the main paper showed only a selection of those.

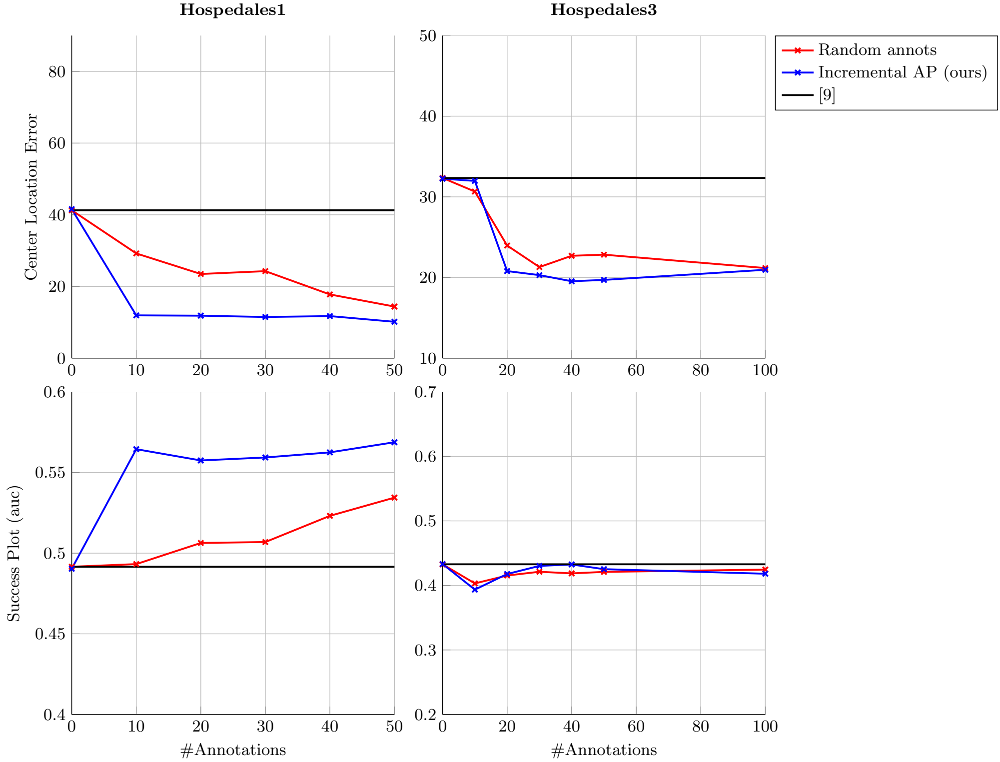

Fig. 1: Annotation selection on both datasets and both metrics (*c.f.* Fig. 7a).

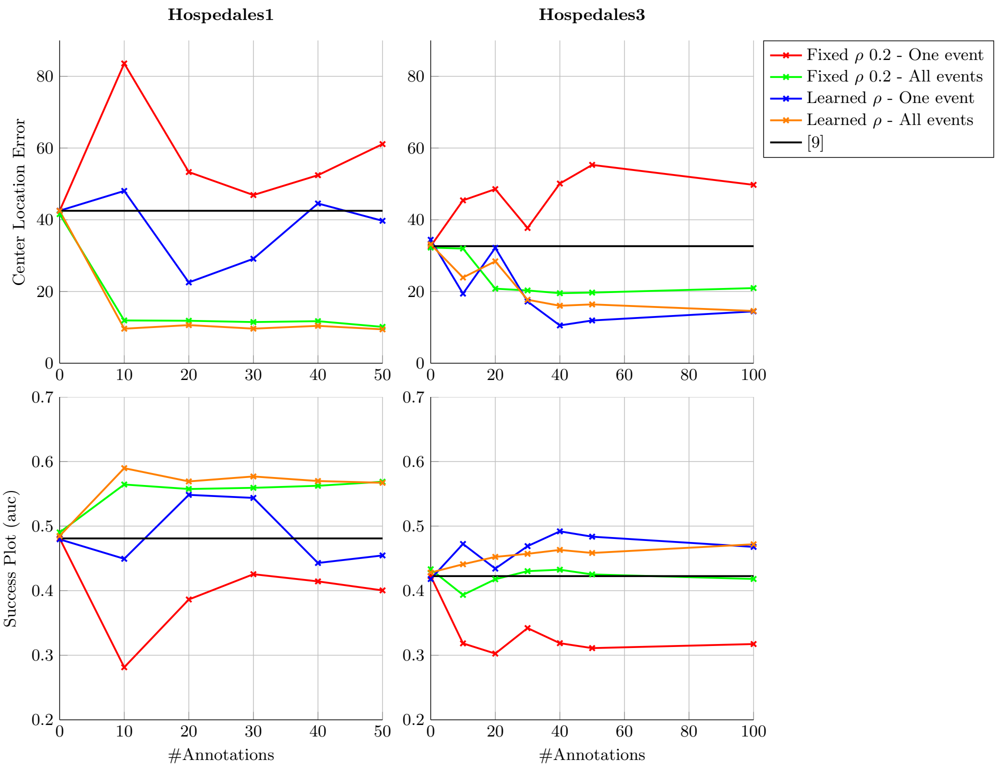

Fig. 2: Event modelling for both datasets and metrics (*c.f.* Fig. 8b).

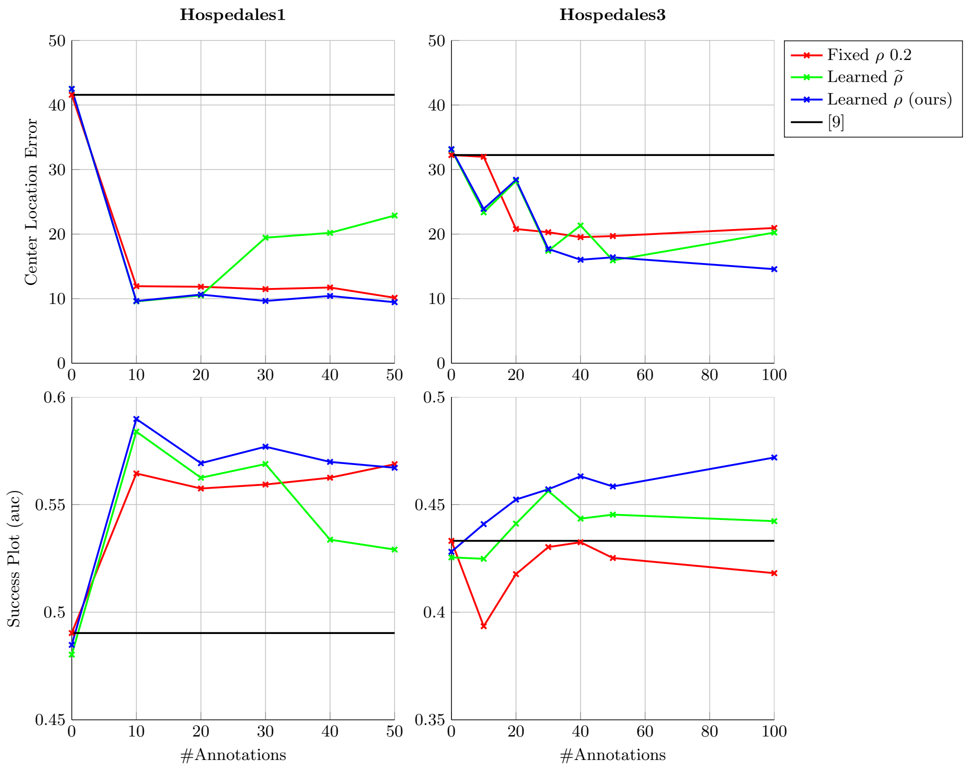

Fig. 3: Learning deceptiveness for both datasets and metrics (*c.f.* Fig. 7b).