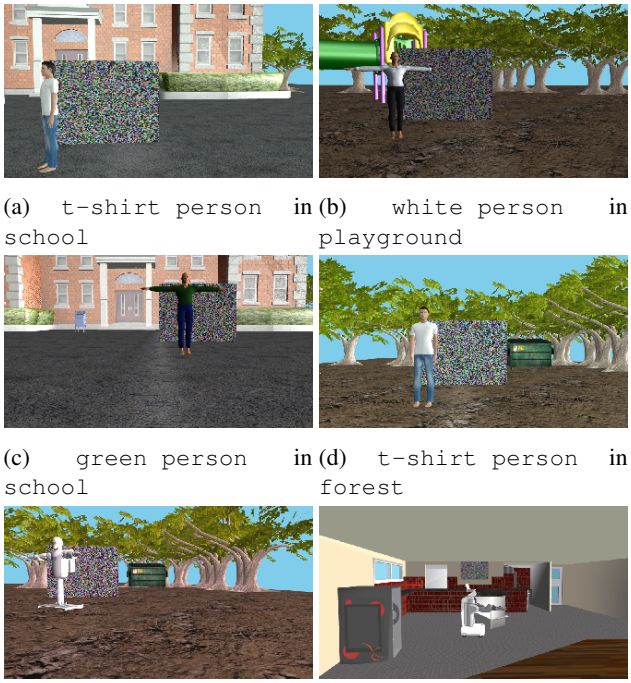
Physical Adversarial Textures That Fool Visual Object Tracking Supplementary Material

Rey Reza Wiyatno Anqi Xu
Element AI
Montreal, Canada
{rey.reza, ax}@elementai.com

1. Simulated Scenarios

Figure 1 depicts samples of the 5 target (human or humanoid robot models) and 4 scenarios (outdoor and indoor scenes) that we created within the Gazebo simulation software [3]. These are used both for generating and evaluating Physical Adversarial Textures (PAT).



(a) `t-shirt person` in `school` (b) `white person` in `playground`

(c) `green person` in `school` (d) `t-shirt person` in `forest`

(e) `robonaut` in `forest` (f) `PR2` in `cafe`

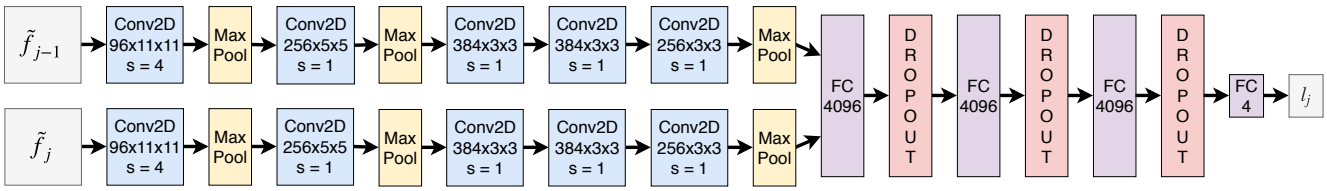
Figure 1: Samples of simulated scenarios.

2. PAT Attack: Random Scene Configuration

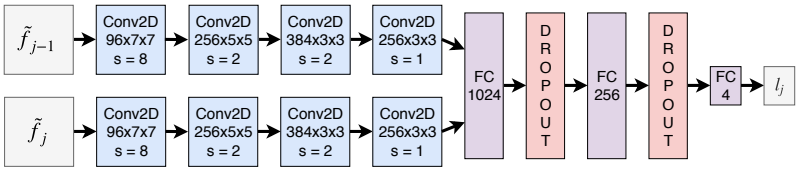
The Expectation Over Transformation (EOT) algorithm [1] randomizes various parameters and aspects of scenes, such as camera placement and target appearance. By optimizing on these diverse and randomized scenes, we can ensure that the generated PAT would likely be universally adversarial. Table 1 presents *default* ranges used for continuous transformation variables used in our PAT Attack process, while Table 2 enumerates selections for discrete transformation variables. This default configuration is used in Sections 6.2, 6.4, and 6.5 in the main paper.

Table 1: Continous EOT variable ranges for PAT attack.

Transformation	Min	Max
Initial camera x (m)	-1.5	1.5
Initial camera y (m)	-11.0	-6.0
Initial camera z (m)	0.6	1.8
Initial camera roll (°)	0.0	0.0
Initial camera pitch (°)	-5.0	5.0
Initial camera yaw (°)	-15.0	15.0
Camera Δx (m)	-0.1	0.1
Camera Δy (m)	-0.5	0.5
Camera Δz (m)	-0.1	0.1
Camera Δroll (°)	0.0	0.0
Camera Δpitch (°)	-3.0	3.0
Camera Δyaw (°)	-3.0	3.0
Initial target x (m)	-1.4	1.4
Initial target y (m)	-5.0	-0.7
Initial target z (m)	0.0	0.0
Initial target roll (°)	0.0	0.0
Initial target pitch (°)	0.0	0.0
Initial target yaw (°)	0.0	180.0
Target Δx (m)	-0.1	0.1
Target Δy (m)	-0.1	0.1
Target Δz (m)	0.0	0.0
Target Δroll (°)	0.0	0.0
Target Δpitch (°)	0.0	0.0
Target Δyaw (°)	-10.0	10.0
Lighting diffuse hue	0.0	360.0
Lighting diffuse saturation	0.0	0.2
Lighting diffuse value	0.1	0.7



(a) Regular-capacity model [2] (`Lg`)



(b) Reduced-capacity model (`Sm`)

Figure 2: Neural architectures for the GOTURN object tracker instances.

Table 2: Discrete EOT variable selections for PAT attack.

Backgrounds	Targets
`school`	`green person`
`forest`	`PR2`

3. Trained GOTURN models

Figure 2 illustrates the two GOTURN neural object tracking architectures used in our experiments.

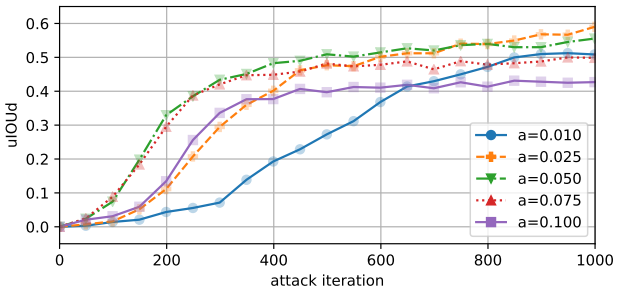
4. Baseline PAT Attack Settings

The parameters used in the baseline PAT attack settings (Section 6.1.3 in the main paper) were determined using hyperparameters search, and from conducting sensitivity analyses on EOT minibatch size and iteration, as well as texture attributes experiments.

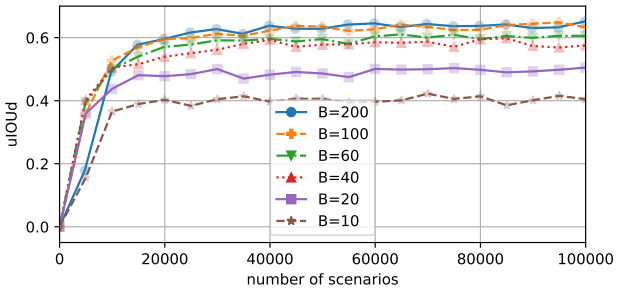
4.1. EOT Minibatch Size and Iteration

Similar to how training a neural network using Stochastic Gradient Descent (SGD) is sensitive to hyperparameter settings, we analyzed the sensitivity of our proposed PAT attack method to its hyperparameters. We suspect that attacks using smaller EOT minibatch sizes B would require more iterations I to converge assuming a fixed perturbation step size α, while attacks using large minibatch sizes B would require an impractical amount of computing time per iteration. Thus, it is practically beneficial to balance the combination of the perturbation step size α and the minibatch size B, given a fixed number of attack iterations I.

We first optimized α for a fixed minibatch size of $B = 20$. As shown in Figure 3a, a step size of $\alpha = 0.025$ attained the best end-performance, however $\alpha = 0.075$ converged initially much faster. This trade-off substantiates our empirical observations and suggests that the source



(a) Perturbation step size α



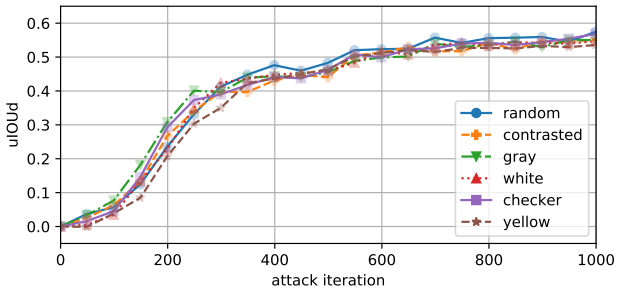
(b) EOT minibatch size B

Figure 3: Adversarial strength over attack iterations, for various α values and EOT minibatch size.

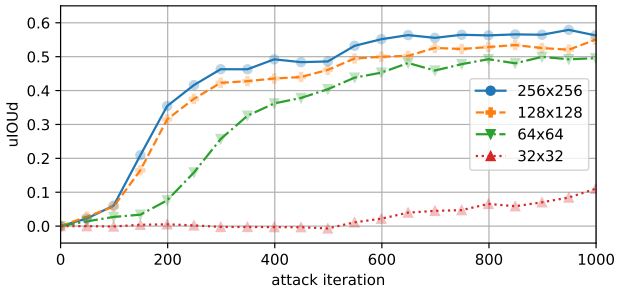
texture initially needs to have most of its pixels *broadly* perturbed to cause adversarial texture patterns to emerge, which would require drastic pixel changes with large perturbation sizes. Subsequently, however, slight *localized* pixel enhancements around "critical adversarial patterns" (see Section 6.4 in the main paper) steadily enhance the PAT's adversarial strength. Thus, we recommend a practical schedule that starts with a large perturbation size of $\alpha = 0.075$ for 500 attack iterations, and then refines using a smaller step size of $\alpha = 0.025$.

Next, using a single non-scheduled perturbation size of $\alpha = 0.075$, we varied the EOT minibatch size B. Note that, in Figure 3b, $\mu IOUd$ is plotted against the number of total EOT scenarios observed, i.e., $B \times I$. These results show consistent performance trends that are proportional to $B \times I$, i.e. the total number of scenes seen by each PAT attack, rather than the number of attack iterations I itself. Also, beyond small values of $B \geq 20$ that lead to high-variance stochastic gradient updates, larger minibatch sizes result in similar and *diminishing* amounts of improvement in both initial convergence speed and asymptotic adversarial strength. Consequently, we chose $B = 20$ for the best trade-off between compute per attack iteration and convergence.

4.2. Texture Attributes



(a) Initial textures



(b) Texture sizes

Figure 4: Adversarial strength among various initial textures and texture sizes.

Various related work made different recommendations on which source texture to use for best results. In particular, suggestions included all-white and all-yellow [5], and a *random contrasted* checkerboard pattern alternating between uniform sampling of $[0, 255]$ and $\{0, 255\}$ [4]. We also tried an all-gray source pattern, as well as a per-pixel randomly-sampled source.

However, as shown in Figure 4a, we found that initializing the texture with different patterns did not result in significant changes in convergence nor performance.

We also explored the effects of changing texture sizes and found that using a resolution of 32×32 lead to consistently poor results, while settings of 64×64, 128×128, and 256×256, yielded little differences in both initial convergence speed and asymptotic performance, as seen in Figure 4b. We thus chose 128×128 to balance between having sufficient pixel capacity to accommodate the wide ranges of EOT conditions, and amount of computation to compute texture perturbations. Still, we found it very important to be aware that our resolution choices are significantly affected by the viewing distances (see Section 2) and poster sizes used in our experiments.

5. Ablation of EOT Conditioning Variables

In Section 6.3 of the main paper, we evaluated the effects of varying the ranges or choices for different EOT transformation variables, including background (`-bg`, `+bg`), target (`-target`, `+target`), lighting (`-light`, `+light`), poster size (`small poster`), camera pose (`-cam pose`, `+cam pose`), and target pose (`-target pose`, `+target pose`). Modified ranges to camera pose, target pose, and lighting are shown in Table 3, 4, and 5, respectively. Also, variations for (`-bg`, `+bg`) and (`-target`, `+target`) are as follows:

- `-bg`: use `playground` only;
- `+bg`: randomize among `school`, `forest`, `playground` and `cafe`;
- `-target`: use `green person` only;
- `+target`: randomize among `green person`, `white person`, `t-shirt person`, `PR2` and `robonaut`.

Table 3: PAT attack settings for `-cam pose` and `+cam pose`.

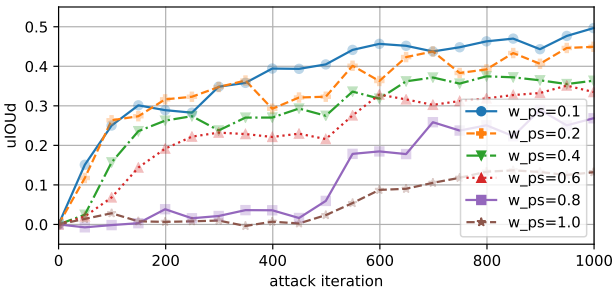
Transformation	`-cam pose`		`+cam pose`	
	Min	Max	Min	Max
Initial x (m)	0.0	0.0	-2.0	2.0
Initial y (m)	-8.5	-8.5	-16.5	-5.5
Initial z (m)	1.2	1.2	0.4	2.2
Initial roll (°)	0.0	0.0	-1.5	1.5
Initial pitch (°)	0.0	0.0	-10.0	10.0
Initial yaw (°)	0.0	0.0	-20.0	20.0
Δx (m)	0.0	0.0	-0.15	0.15
Δy (m)	0.0	0.0	-0.80	0.80
Δz (m)	0.0	0.0	-0.15	0.15
Δroll (°)	0.0	0.0	0.0	0.0
Δpitch (°)	0.0	0.0	-5.0	5.0
Δyaw (°)	0.0	0.0	-5.0	5.0

Table 4: PAT attack settings for `-target pose` and `+target pose`.

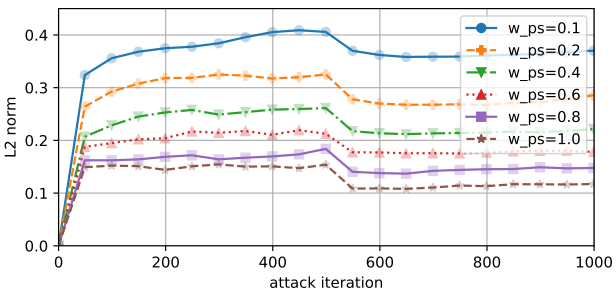
Transformation	`-target pose`		`+target pose`	
	Min	Max	Min	Max
Initial x (m)	0.0	0.0	-1.6	1.6
Initial y (m)	-2.7	-2.7	-5.0	-0.7
Initial z (m)	0.0	0.0	0.0	0.0
Initial roll (°)	0.0	0.0	0.0	0.0
Initial pitch (°)	0.0	0.0	0.0	0.0
Initial yaw (°)	90.0	90.0	-90.0	270.0
Δx (m)	0.0	0.0	-0.15	0.15
Δy (m)	0.0	0.0	-0.15	0.15
Δz (m)	0.0	0.0	0.0	0.0
Δroll (°)	0.0	0.0	0.0	0.0
Δpitch (°)	0.0	0.0	0.0	0.0
Δyaw (°)	0.0	0.0	-20.0	20.0

Table 5: PAT attack settings for `-light` and `+light`.

Diffuse Light Source	`-light`		`+light`	
	Min	Max	Min	Max
Hue	0.0	360.0	0.0	360.0
Saturation	0.0	0.0	0.0	0.7
Value	0.7	0.7	0.0	0.7



(a) uIOUd



(b) L_2-norm perceptual similarity

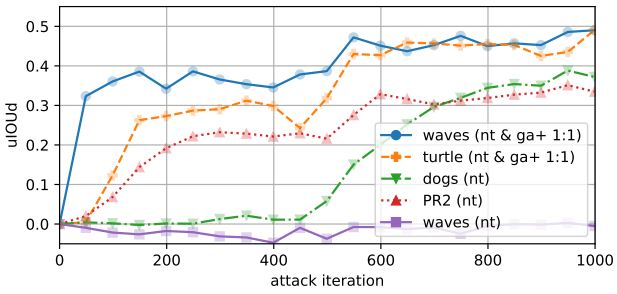
Figure 5: Adversarial strength and perceptual similarity among various w_{ps}.

6. Imitation Attacks

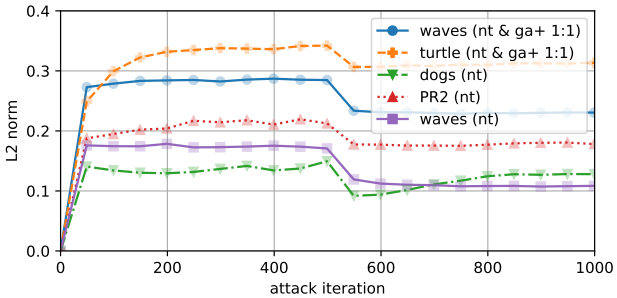
In Section 6.4 of the main paper, we set the value of $w_{ps} = 0.6$. This value was determined based on an experiment where we studied the effect of changing w_{ps} on the adversarial strength $\mu IOUd$ and perceptual similarity (as measured by the Euclidean L_2 distance to the source image in RGB colorspace). Unsurprisingly, as seen in Figure 5, smaller values of w_{ps} imposed fewer constraints and thus lead to faster attack convergence and better end-performance, while the inverse was true for larger values of w_{ps}. We thus chose $w_{ps} = 0.6$ after manually assessing which PATs had recognizable levels of perceptual similarity to their source images, as seen in Figure 10.

To substantiate Figure 6 in the main paper, Figure 6 (in this document) illustrates how $\mu IOUd$ and perceptual similarity metrics change over attack iterations. As we can see, some specific combinations of initial posters and losses made the attack easier to converge. For example, performing attacks using `waves` as initial texture with hybrid losses ($\mathcal{L}_{nt}$ & $\mathcal{L}_{ga+}$) resulted in strong adversaries, while using non-targeted loss alone did not.

Figure 7 compares perceptual similarity ($L_2\,norm$) against adversarial strength ($\mu IOUd$) among $\mathcal{L}_{ga-}$, $\mathcal{L}_{nt}$, and $\mathcal{L}_{t-}$, and shows that, for a given threshold on $L_2\,norm$, $\mathcal{L}_{ga-}$ generally had better early convergence, but likely

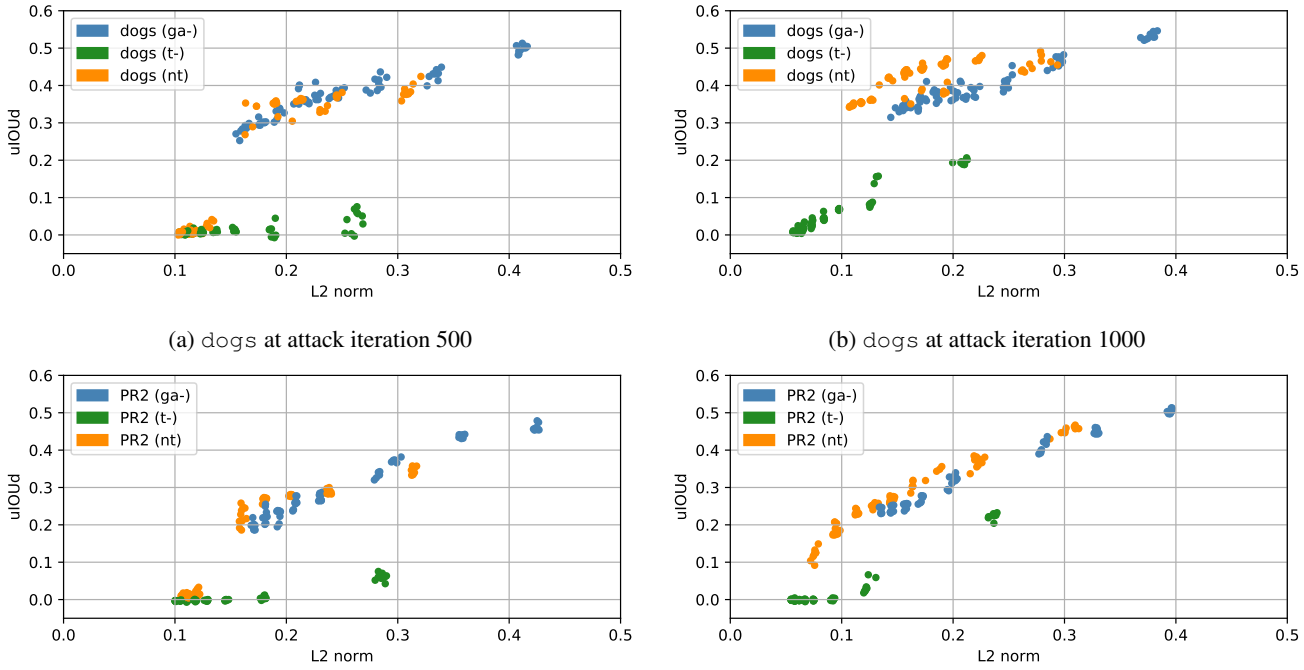


(a) uIOUd



(b) L_2-norm perceptual similarity

Figure 6: Adversarial strength and perceptual similarity among source textures shown in Figure 6 of main paper.



(a) `dogs` at attack iteration 500

(b) `dogs` at attack iteration 1000

(c) `PR2` at attack iteration 500

(d) `PR2` at attack iteration 1000

Figure 7: Performance of $\mathcal{L}_{nt}$, $\mathcal{L}_{ga-}$, and $\mathcal{L}_{t-}$ in imitating `dogs` and `PR2` textures for $w_{ps} \in [0.1 : 0.1 : 0.8]$.

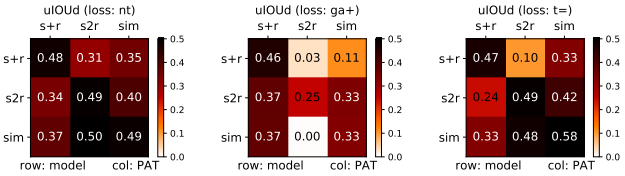
weakened adversarial strength after more attack iterations.

Figure 11 illustrates the emergence of "critical adversarial patterns" that we discussed in Section 6.4 in the main paper. In Figure 11a, the *critical* dark striped pattern started to emerge at around iteration 400, followed by the appearances of other nearby colorful patterns, which presumably were to drive predictions towards the central adversarial striped pattern. In contrast, when we imposed a perceptual similarity loss during an imitation attack, only the dark striped pattern eventually emerged after significantly more attack iterations, as seen in Figure 11b.

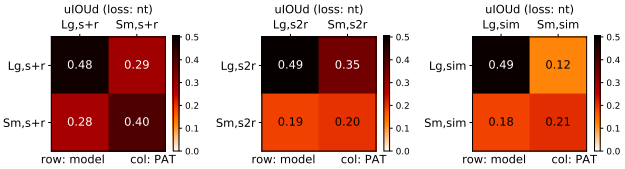
7. Transfer among tracking models

We evaluated the transferability of PATs among different tracking models. When evaluating PATs on GOTURN models trained using different datasets, the off-diagonal results in Figure 8a generally show that a decent-to-great amount of adversarial strength is still present. Nevertheless, we see that the transferred efficacy of adversaries varied based on the tracker model and the loss used. For instance, a `sim`-trained PAT optimized using $\mathcal{L}_{nt}$ and applied to the `s2r` GOTURN tracker is strongly adversarial, whereas a similar PAT optimized using $\mathcal{L}_{ga+}$ becomes completely inert.

Similarly, PATs preserved some of their adversarial strength when transferred between trackers with different capacities, as seen in Figure 8b. However, while all PATs applied to reduced-capacity models (`Sm`) affected GOTURN predictions, their $\mu IOUd$ values around 0.20 do not reflect strong adversaries, thus indicating that it is more difficult to fool small-capacity GOTURN networks into consistently breaking away from their intended target.



(a) Sim&real (`s+r`), sim-to-real (`s2r`), `sim`-only trained models



(b) Models with default (`Lg`) and reduced (`Sm`) capacities

Figure 8: Adversarial strength of generated PATs (columns) applied to different GOTURN tracking models (rows).

Figure 9 shows the PATs used in this experiment. Generally, we observe similar adversarial patterns emerging from PAT Attacks on GOTURN models trained on different

datasets, as well as different capacities, which explain why PATs transfer to a certain degree among different GOTURN trackers. The sole exception is seen from the second row of Figure 9a, which reflected the fact that the adversarial loss $\mathcal{L}_{ga+}$ caused different patterns to emerge for different models, albeit with similar levels of competent adversarial strength.

8. Demonstration of Sim-to-real Transfer

As discussed in Section 6.5 in the main paper, we conducted test runs in real-world tracking and servoing conditions, and qualitatively verified the transferred adversarial strength of our synthetically-generated PATs, especially those containing "critical adversarial patterns". Many of these real-world runs are shown in the supplementary video. Nevertheless, it is generally difficult to quantify performance consistently in the real world, due to tediousness and impracticality in labeling performance, controlling for repeated conditions, and dealing with practical complexities such as limited battery life and hardware failures. Still, we segmented runs into video clips, and manually labeled them as either `strong`ly adversarial (i.e. where the tracker jumps onto the PAT and stays locked onto it even when momentarily obstructed), `weak`ly adversarial (i.e. where the tracker sometimes switches from the person to the PAT, and tends to latch back onto the person), or `fail`ure.

Looking at Table 6, we see that the tracker was quickly drawn to PATs when deployed on a stationary camera. On the other hand, it was much harder to fool the person tracker when the drone was servoing the target. Whether the PAT was displayed digitally on a monitor, or printed as an `A0` poster, we anecdotally observed that both of these materials displayed some amount of specular reflections. These specularities changed as the camera moved around, and thus likely had altered the appearances of PATs during our servoing runs and rendered them inert. Therefore, devising adversaries that are robust to specularities would be an exciting avenue for future research.

Table 6: Physical-world attack performance.

Runs	Strong	Weak	Fail
Stationary	57 (71%)	13 (16%)	10 (13%)
Servo	6 (33%)	5 (28%)	7 (39%)

References

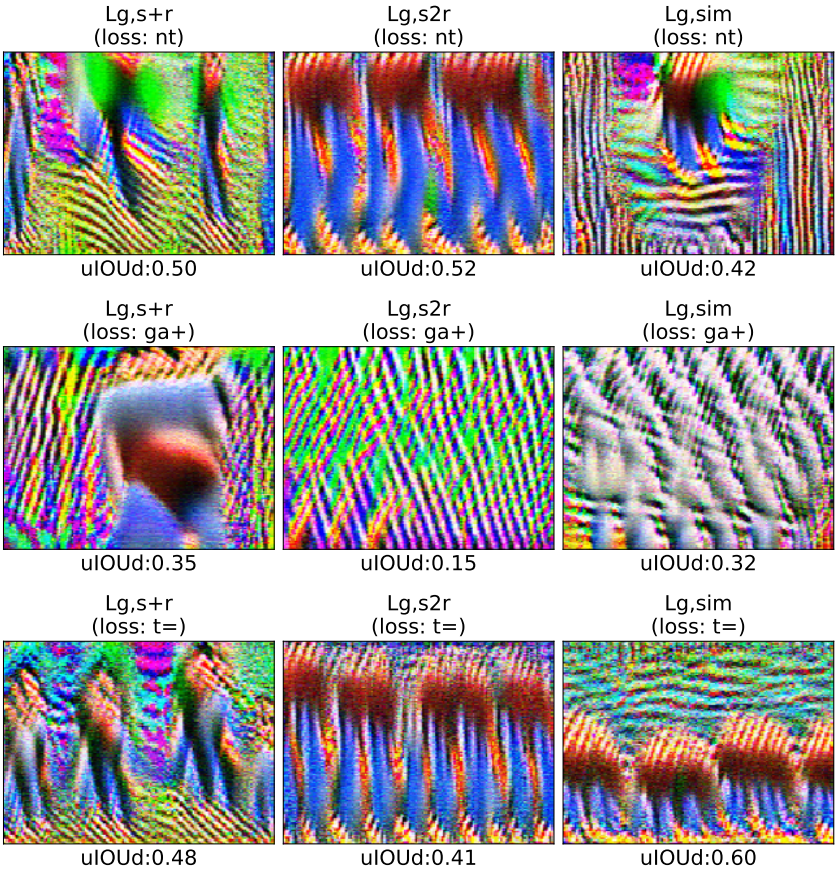
[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *proceedings of the 35th International Conference on Machine Learning (ICML)*, Sweden, 2018.

[2] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

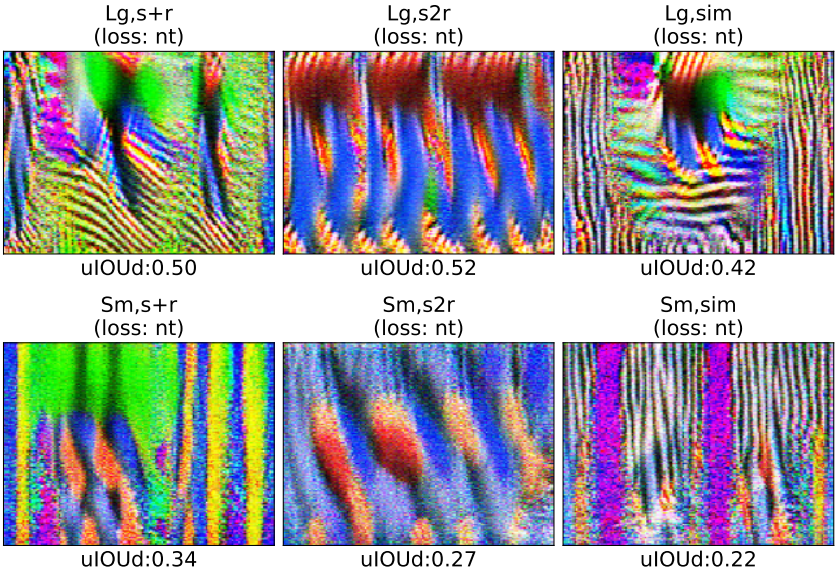
[3] Nathan P. Koenig and Andrew Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[4] Yang Zhang, Hassan Foroosh, Philip David, and Boqing Gong. CAMOU: Learning physical vehicle camouflages to adversarially attack detectors in the wild. In *proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[5] Husheng Zhou, Wei Li, Yuankun Zhu, Yuqun Zhang, Bei Yu, Lingming Zhang, and Cong Liu. Deepbillboard: Systematic physical-world testing of autonomous driving systems. *CoRR*, abs/1812.10812, 2018.



(a) Different training datasets for GOTURN models



(b) Different network capacities for GOTURN models

Figure 9: PATs used in the transferability among tracking models experiment.

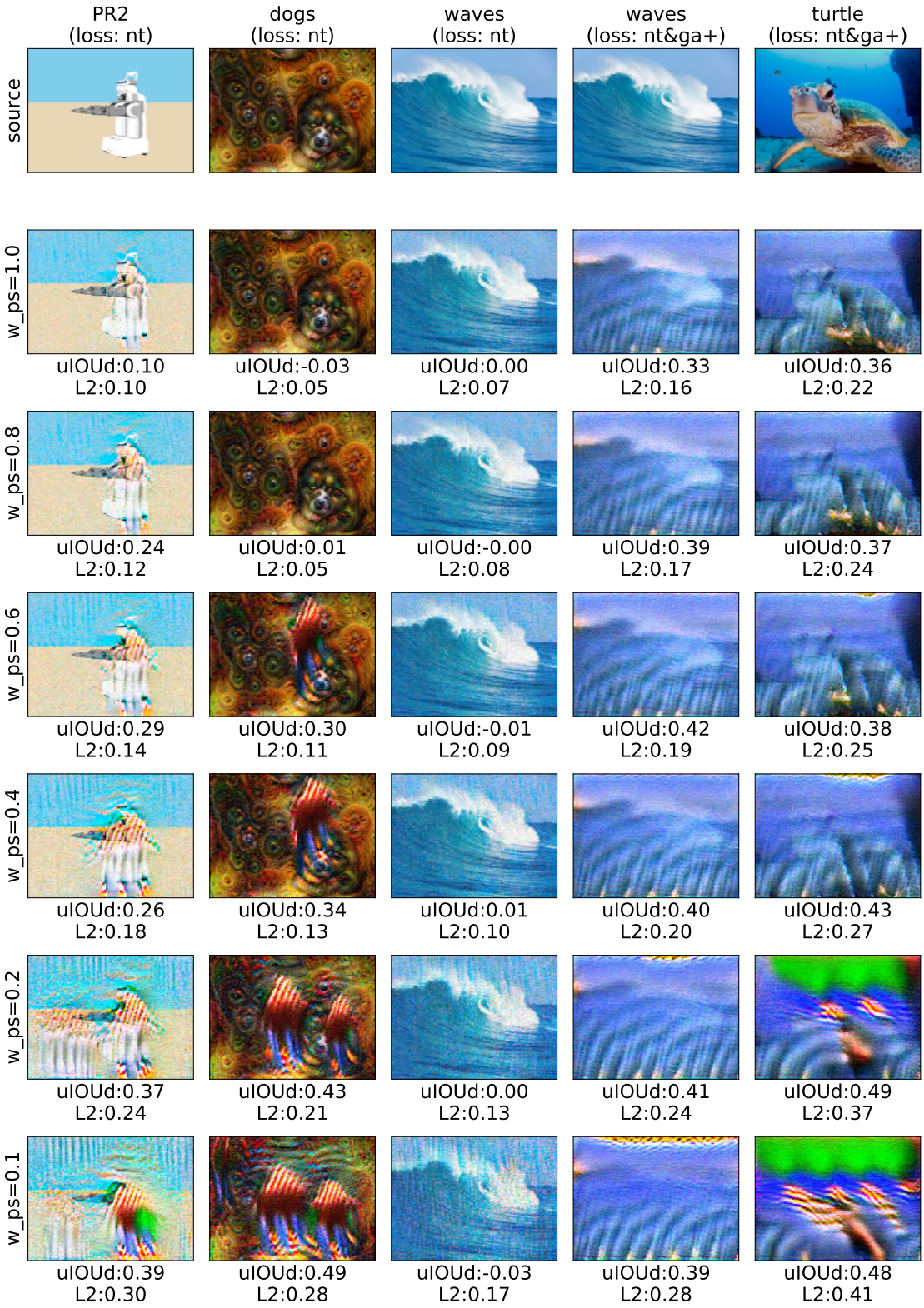
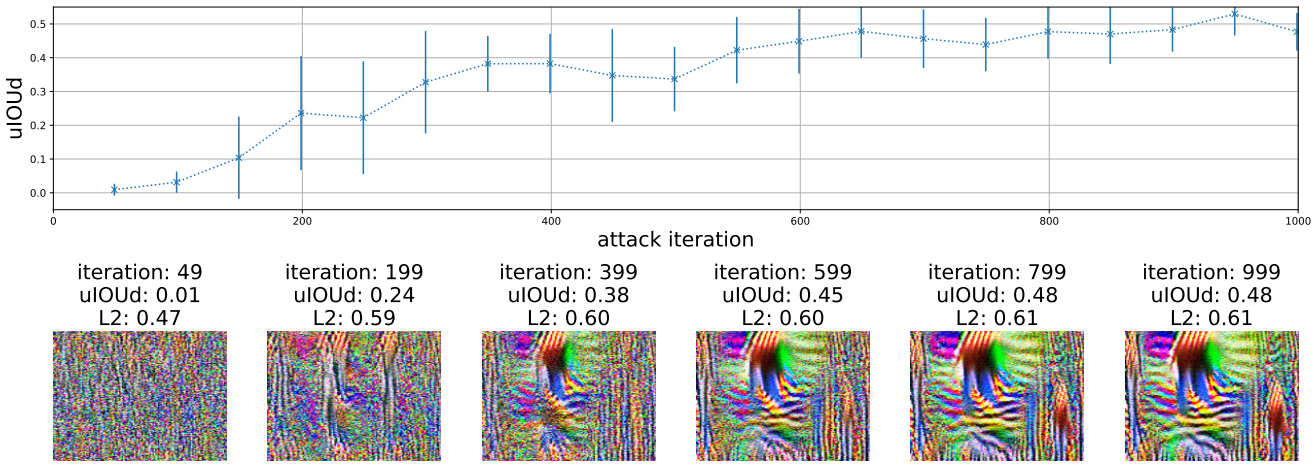
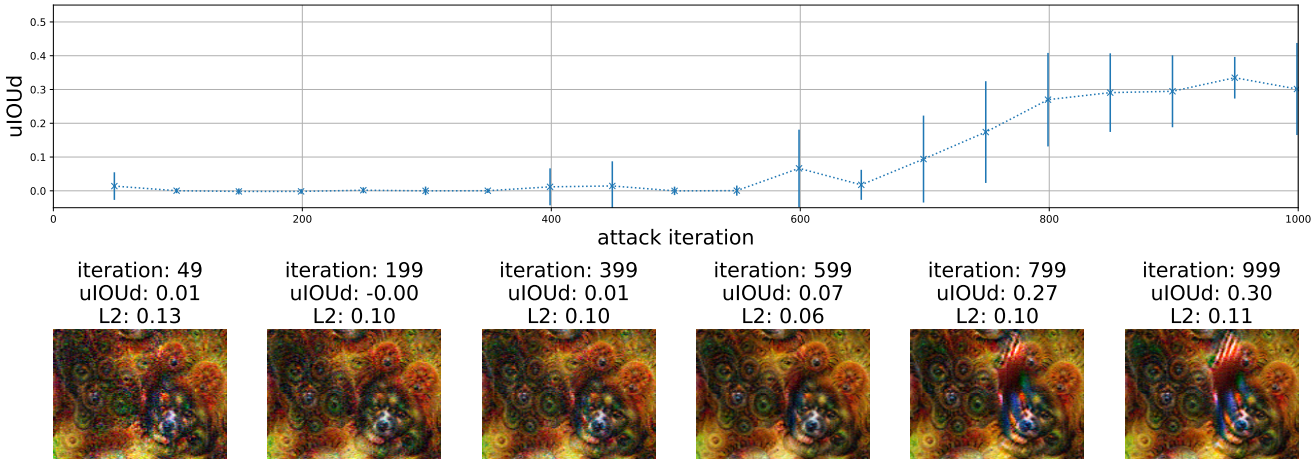


Figure 10: PATs generated with various w_{ps} values.



(a) Non-imitation attack



(b) Imitation attack

Figure 11: The emergence of "critical adversarial patterns" for non-imitation and imitation attacks.

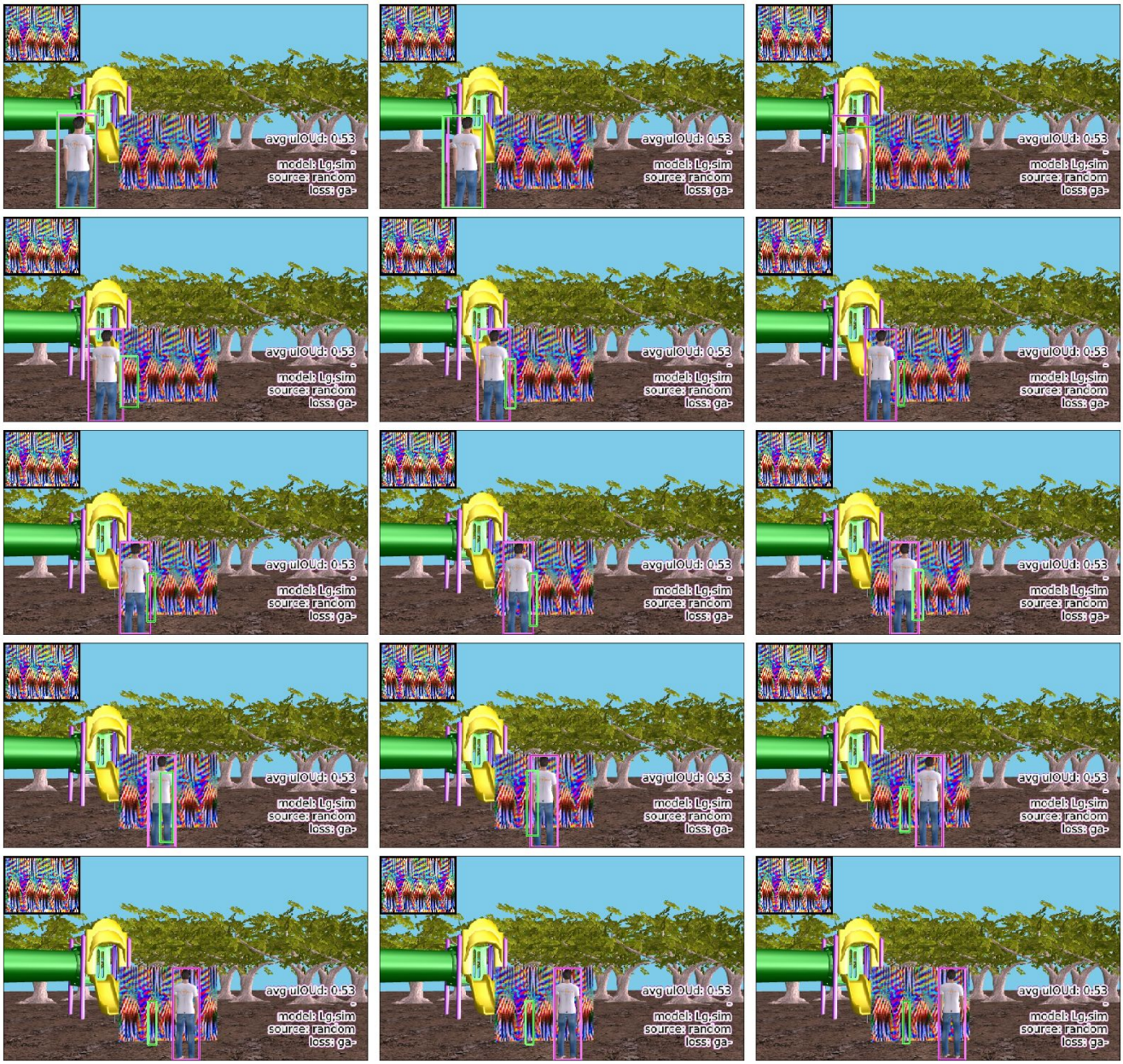


Figure 12: PAT fools the tracker in simulation. Here, the purple bounding box represents the ground truth bounding box of the tracked object, while the green bounding box represents the tracker's prediction. Note that the sequence starts from the top-left frame to the bottom-right frame.

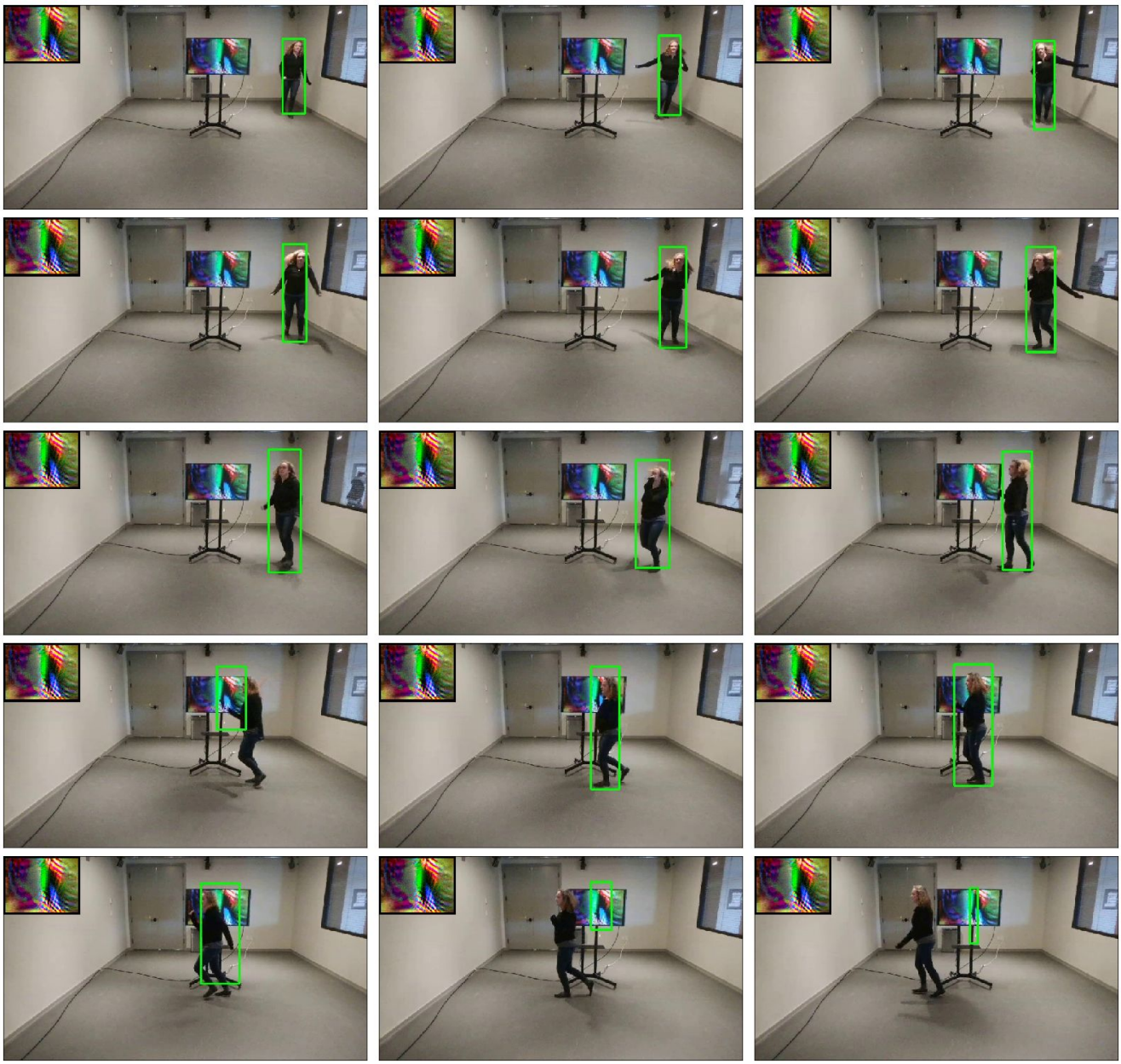


Figure 13: PAT fools the tracker in the real world indoor setting, where the PAT is displayed on a TV. Note that the sequence starts from the top-left frame to the bottom-right frame.

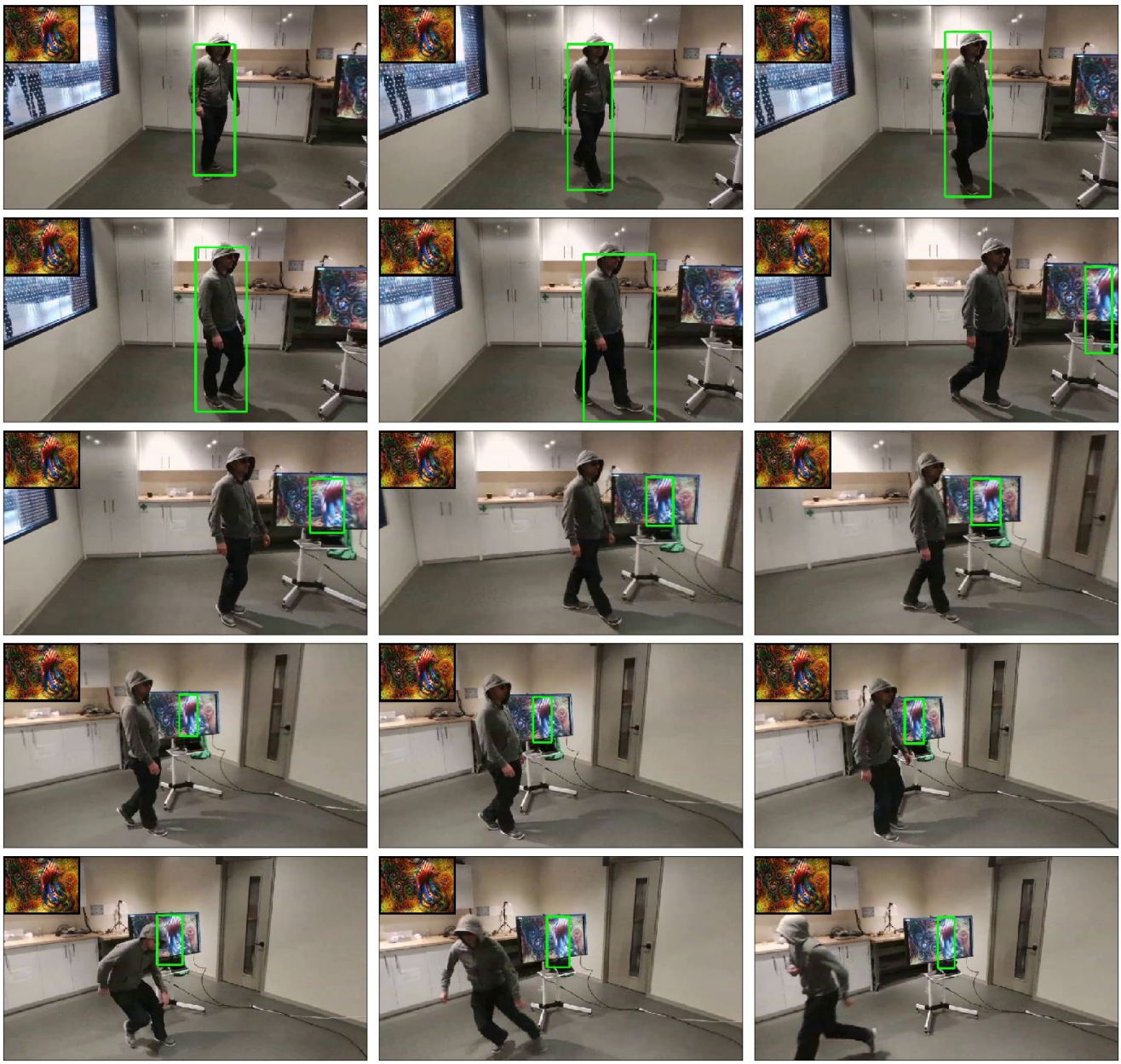


Figure 14: PAT fools the tracker in the real world indoor setting during servoing run. Note that the sequence starts from the top-left frame to the bottom-right frame.

Figure 15: PAT fools the tracker in the real world outdoor setting during servoing run, where the PAT is printed as a poster. Note that the sequence starts from the top-left frame to the bottom-right frame.