A Adversarial attacks algorithms

We present algorithms for both our PGD (Algorithm 1) and universal (Algorithm 2) attacks. In both cases we make use of the PGD adversarial attack scheme [21] to optimize a single adversarial patch. In each optimization step, we update the patch based on the gradient of the training criterion. Finally, we return the produced patch which maximized the evaluation criterion.

Algorithm 1 PGD adversarial attack

Input VO: VO model
Input A: Adversarial patch perturbation
Input (x, y): Trajectory to attack and it's ground truth motions
Input $(\ell_{train}, \ell_{eval})$: Train and evaluation loss functions
Input α: Step size for the attack

$P \leftarrow \text{Uniform}(0, 1)$
$P_{\text{best}} \leftarrow P$
$\text{Loss}_{\text{best}} \leftarrow 0$
for $k = 1$ to K **do**
  optimization step:
  $g \leftarrow \nabla_P \ell_{train}(VO(A(x, P)), y)$
  $P \leftarrow P + \alpha \cdot \text{sign}(g)$
  $P \leftarrow clip(P, 0, 1)$
  evaluate patch:
  $\text{Loss} \leftarrow \ell_{eval}(VO(A(x, P)), y)$
  if $\text{Loss} > \text{Loss}_{\text{best}}$ **then**
    $P_{\text{best}} \leftarrow P$
    $\text{Loss}_{\text{best}} \leftarrow \text{Loss}$
  end if
end for
return P_{best}

B Real data experiment specifics

For the generation of the real dataset, in addition to the Mocap markers we make use of Aruco markers to produce the patch's coordinates in the camera system for each frame, which are then used for generating the patch's mask. In addition, the Aruco Markers, being printed on paper or some other material, provide an estimate for the albedo extremes of the printed patch on the same material. In each frame, we than make use of the detected patch to estimate its albedo limits. We calculate these limits by fitting the pixel histogram of the patch area to a Bivariate normal distribution. We account for the illumination variation within this area by multiplying our albedo images by the lightness channel of the HSL (hue, saturation, lightness) representation of the original image. As seen

Algorithm 2 Universal PGD adversarial attack

Input VO: VO model
Input A: Adversarial patch perturbation
Input (X_{train}, Y_{train}): Trajectories training dataset
Input (X_{eval}, Y_{eval}): Trajectories evaluation dataset
Input $(\ell_{train}, \ell_{eval})$: Training and evaluation loss functions
Input (N_{train}, N_{eval}): Number of training and evaluation trajectories
Input α: Step size for the attack

$P \leftarrow \text{Uniform}(0, 1)$
$P_{\text{best}} \leftarrow P$
$\text{Loss}_{\text{best}} \leftarrow 0$
for $k = 1$ to K **do**
    optimization step:
    $g \leftarrow 0$
    for $i = 1$ to N_{train} **do**
        $\hat{y}_{train,i} \leftarrow VO(A(x_{train,i}, P))$
        $g \leftarrow g + \nabla_P \ell_{train}(\hat{y}_{train,i}, y_{train,i})$
    end for
    $P \leftarrow P + \alpha \cdot \text{sign}(g)$
    $P \leftarrow clip(P, 0, 1)$
    evaluate patch:
    $\text{Loss} \leftarrow 0$
    for $i = 1$ to N_{eval} **do**
        $\hat{y}_{eval,i} \leftarrow VO(A(x_{eval,i}, P))$
        $\text{Loss} \leftarrow \text{Loss} + \ell_{eval}(\hat{y}_{eval,i}, y_{eval,i})$
    end for
    if $\text{Loss} > \text{Loss}_{\text{best}}$ **then**
        $P_{\text{best}} \leftarrow P$
        $\text{Loss}_{\text{best}} \leftarrow \text{Loss}$
    end if
end for
return P_{best}

in Fig. 3, the black and white albedo images accordingly resemble the black and white pixels in the Aruco markers.

Throughout the data-set generation process, we discarded trajectories with incomplete camera pose or patch coordinates.

The trajectories initial positions formed an horizontal angle range of $[-8.5°, 8.5°]$ with respect to the target patch plane.