Lipschitz Optimization for Formal Verification of Homographies

Supplementary Material

7. Notation

$\triangleq$	Equality by definition
$[\![1, n]\!]$	Set of natural integers between 1 and n, both included
$\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$	Sets (cursive script)
Diff $(\mathcal{B})$	Subset of $\mathcal{B}$ where the bound error $\underline{f}$ is differentiable
f^*	Optimum value (maximum or minimum) of function f
$\widehat{f^*}$	Estimate of f^*
$G_{i,j}^{\boldsymbol{x_0}}(\boldsymbol{\kappa})$	Value of pixel (i, j) in the transformed image, when image $\boldsymbol{x_0}$ is transformed by $\boldsymbol{\kappa}$
$\boldsymbol{x}$	Vectors, with 1-indexed elements x_i (bold lowercase)
$\boldsymbol{x}^A$	Vector $\boldsymbol{x}$, expressed in coordinate frame A
$\boldsymbol{A}$	Matrix, with 1-indexed elements $A_{i,j}$ (bold uppercase)
(A)	The (A) coordinate frame
$\boldsymbol{R}_A^B$	3×3 passive rotation from A-frame to B-frame coordinates, so that $\boldsymbol{x}^B = \boldsymbol{R}_A^B \boldsymbol{x}^A$
$\boldsymbol{t}_{B\to A}^B$	3×1 translation vector from B-frame to A-frame, expressed in frame B
$\boldsymbol{T}_A^B$	4×4 passive transform from (A) to (B), so $\boldsymbol{x}^B = \boldsymbol{T}_A^B \boldsymbol{x}^A$. Note $\boldsymbol{T}_A^B = \left[\boldsymbol{R}_A^B \middle\vert \boldsymbol{t}_{B\to A}^B\right]$
$[\boldsymbol{R}\vert\boldsymbol{t}]$	Simplified form of the 4×4 homogeneous coordinate transform $\left[\begin{array}{ccc\vert c} & \boldsymbol{R} & & \boldsymbol{t} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$

8. Datasets

Figure 6. Sample images from each dataset.

9. Padding

Geometric transformations frequently map target coordinates beyond the boundaries of the source image. In this case, bilinear interpolation relies on padding mechanisms to impute missing pixel values. Depending on the application, different padding

techniques exist, with varying levels of complexity and realism. For the traffic sign of Fig. 7, "reflect" padding offers a perceptually plausible extension of the background texture. Note how padding introduces non-existent signal into the image, which is distinct from having a true change of camera perspective with for example, occlusion effects.

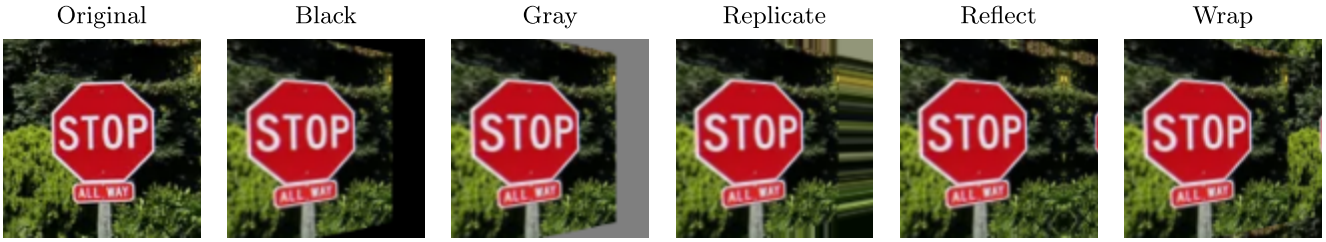

Figure 7. Overview of different padding techniques.

10. Transforms glossary

We detail a few scenarios that simplify the general homography form given in Eq. (7). Under our assumptions, we show that some of these scenarios simplify into exact affine transforms. We also notice that rotation perturbations are independent of the pose of the initial viewpoint. For brevity of the translation cases, we simplify further by assuming a camera initially aligned with the world axes ($\phi, \theta, \psi = 0$). We keep $x, y, z \neq 0$, and notice that the homography does not depend on the initial camera longitudinal or lateral position. It depends however on the delta in position $\Delta x, \Delta y, \Delta z$. To explicitly highlight the effects of the transform on the image, our figures utilize black padding. However, we refer to Sec. 9 for padding techniques that prioritize perceptual realism. The more complete forms are available with our public code.

10.1. Roll perturbation $\boldsymbol{\Delta\phi}$

Figure 8. Samples for a roll perturbation, with $\boldsymbol{\kappa} = \Delta\phi \in \mathcal{B} = [0, 15]^\circ$

We notice that under our assumptions, the roll perturbation is an affine transform and corresponds exactly to a 2D rotation of the image. It is also independent of the camera focal length f.

$$\boldsymbol{H}^{-1}(\Delta\phi) = \begin{bmatrix} \cos(\Delta\phi) & -\sin(\Delta\phi) & -x_c\cos(\Delta\phi) + x_c + y_c\sin(\Delta\phi) \\ \sin(\Delta\phi) & \cos(\Delta\phi) & -x_c\sin(\Delta\phi) - y_c\cos(\Delta\phi) + y_c \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$\Big(u_0(\Delta\phi), v_0(\Delta\phi)\Big) = \begin{pmatrix} (u - x_c)\cos(\Delta\phi) - (v - y_c)\sin(\Delta\phi), \\ (u - x_c)\sin(\Delta\phi) + (v - x_c)\cos(\Delta\phi) \end{pmatrix} \tag{16}$$

$$\nabla^\top_{\Delta\phi} T^{-1}(\Delta\phi) = \begin{pmatrix} -(u - x_c)\sin(\Delta\phi) - (v - y_c)\cos(\Delta\phi), \\ (u - x_c)\cos(\Delta\phi) - (v - y_c)\sin(\Delta\phi) \end{pmatrix} \tag{17}$$

$$\mathrm{Diff}(\mathcal{B}) = \mathcal{B} \tag{18}$$

$$\underset{\Delta\phi \in \mathrm{Diff}(\mathcal{B})}{\arg\max} \Big|\nabla_{\Delta\phi} J\left(\Delta\phi\right)\Big| \in \left\{ \Delta\phi_{\min}, \Delta\phi_{\max}, \arctan\left(-\frac{v - y_c}{u - x_c}\right), \arctan\left(\frac{u - x_c}{v - y_c}\right) \right\} \pmod{\pi} \tag{19}$$

The limit cases of $\arctan$ apply when $u = x_c$ and $v = y_c$.

10.2. Pitch perturbation $\Delta\theta$

Figure 9. Samples for a pitch perturbation, with $\boldsymbol{\kappa} = \Delta\theta \in \mathcal{B} = [0, 15]^\circ$

$$\boldsymbol{H}^{-1}(\Delta\theta) = \begin{bmatrix} 1 & \frac{x_c \sin(\Delta\theta)}{f} & \frac{x_c(f\cos(\Delta\theta) - f - y_c \sin(\Delta\theta))}{f} \\ 0 & \cos(\Delta\theta) + \frac{y_c \sin(\Delta\theta)}{f} & -\frac{\left(f^2 + y_c^2\right)\sin(\Delta\theta)}{f} \\ 0 & \frac{\sin(\Delta\theta)}{f} & \cos(\Delta\theta) - \frac{y_c \sin(\Delta\theta)}{f} \end{bmatrix} \tag{20}$$

$$\Big(u_0(\Delta\theta), v_0(\Delta\theta)\Big) = \Big(x_c + f \cdot \tfrac{u - x_c}{f\cos(\Delta\theta) + (v - y_c)\sin(\Delta\theta)},\ y_c - f \cdot \tfrac{f\sin(\Delta\theta) - (v - y_c)\cos(\Delta\theta)}{f\cos(\Delta\theta) + (v - y_c)\sin(\Delta\theta)}\Big) \tag{21}$$

$$\nabla_{\Delta\theta}^{\top} T^{-1}(\Delta\theta) = \Big(-\tfrac{f(u - x_c)[-f\sin(\Delta\theta) + (v - y_c)\cos(\Delta\theta)]}{[f\cos(\Delta\theta) + (v - y_c)\sin(\Delta\theta)]^2},\ \tfrac{-f\left[f^2 + (v - y_c)^2\right]}{[f\cos(\Delta\theta) + (v - y_c)\sin(\Delta\theta)]^2}\Big) \tag{22}$$

$$\mathrm{Diff}(\mathcal{B}) = \mathcal{B} \setminus \{\Delta\theta_c \pmod{\pi}\} \qquad \text{with } \Delta\theta_c = \arctan\left(-\frac{f}{v - y_c}\right) \tag{23}$$

$$\underset{\Delta\theta \in \mathrm{Diff}(\mathcal{B})}{\arg\max} \left|\nabla_{\Delta\theta} J\left(\Delta\theta\right)\right| \in \left\{\Delta\theta_{\min},\ \Delta\theta_{\max},\ \arctan\left(\frac{v - y_c}{f}\right) \pmod{\pi}\right\} \tag{24}$$

10.3. Yaw perturbation $\Delta\psi$

Figure 10. Samples for a yaw perturbation, with $\boldsymbol{\kappa} = \Delta\psi \in \mathcal{B} = [0, 15]^\circ$

$$\boldsymbol{H}^{-1}(\Delta\psi) = \begin{bmatrix} \cos(\Delta\psi) - \frac{x_c \sin(\Delta\psi)}{f} & 0 & \frac{\left(f^2 + x_c^2\right)\sin(\Delta\psi)}{f} \\ -\frac{y_c \sin(\Delta\psi)}{f} & 1 & \frac{y_c(f(\cos(\Delta\psi) - 1) + x_c \sin(\Delta\psi))}{f} \\ -\frac{\sin(\Delta\psi)}{f} & 0 & \cos(\Delta\psi) + \frac{x_c \sin(\Delta\psi)}{f} \end{bmatrix} \tag{25}$$

$$\Big(u_0(\Delta\psi), v_0(\Delta\psi)\Big) = \Big(x_c + f \cdot \tfrac{f\sin(\Delta\psi) + (u - x_c)\cos(\Delta\psi)}{f\cos(\Delta\psi) - (u - x_c)\sin(\Delta\psi)},\ y_c + f \cdot \tfrac{v - y_c}{f\cos(\Delta\psi) - (u - x_c)\sin(\Delta\psi)}\Big) \tag{26}$$

$$\nabla_{\Delta\psi}^{\top} T^{-1}(\Delta\psi) = \Big(\tfrac{f\left[f^2 + (u - x_c)^2\right]}{[f\cos(\Delta\psi) - (u - x_c)\sin(\Delta\psi)]^2},\ \tfrac{f(v - y_c)[f\sin(\Delta\psi) + (u - x_c)\cos(\Delta\psi)]}{[f\cos(\Delta\psi) - (u - x_c)\sin(\Delta\psi)]^2}\Big) \tag{27}$$

$$\mathrm{Diff}(\mathcal{B}) = \mathcal{B} \setminus \{\Delta\psi_c \pmod{\pi}\} \qquad \text{with } \psi_c = \arctan\left(\frac{f}{u - x_c}\right) \tag{28}$$

$$\underset{\Delta\psi \in \mathrm{Diff}(\mathcal{B})}{\arg\max} \left|\nabla_{\Delta\psi} J\left(\Delta\psi\right)\right| \in \left\{\Delta\psi_{\min},\ \Delta\psi_{\max},\ \arctan\left(-\frac{u - x_c}{f}\right) \pmod{\pi}\right\} \tag{29}$$

10.4. Translation perturbation Δx

Figure 11. Samples for a x-translation perturbation, with $\boldsymbol{\kappa} = \Delta x \in \mathcal{B} = [0, 1]m$

We notice here that moving the camera forward does not exactly match a simple zooming operation on the image, and is a more involved transform.

$$\boldsymbol{H}^{-1}(\Delta x) = \begin{bmatrix} 1 & -\frac{\Delta x x_c}{fz} & \frac{\Delta x x_c y_c}{fz} \\ 0 & 1 - \frac{\Delta x y_c}{fz} & \frac{\Delta x y_c^2}{fz} \\ 0 & -\frac{\Delta x}{fz} & 1 + \frac{\Delta x y_c}{fz} \end{bmatrix} \quad \text{with } z \neq 0 \text{ (i.e. camera not on ground plane)} \tag{30}$$

$$\Big(u_0(\Delta x), v_0(\Delta x)\Big) = \left(\frac{\Delta x(v-y_c)x_c - fzu}{\Delta x(v-y_c) - fz}, \quad \frac{\Delta x(v-y_c)y_c - fzv}{\Delta x(v-y_c) - fz}\right) \tag{31}$$

$$\nabla_{\Delta x}^{\top} T^{-1}(\Delta x) = \left(\frac{fz(u-x_c)(v-y_c)}{[\Delta x(v-y_c) - fz]^2}, \quad \frac{fz(v-y_c)^2}{[\Delta x(v-y_c) - fz]^2}\right) \tag{32}$$

$$\text{Diff}(\mathcal{B}) = \mathcal{B} \setminus \left\{ \frac{fz}{v - y_c} \right\} \quad \text{when } v \neq y_c, \mathcal{B} \text{ otherwise} \tag{33}$$

$$\underset{\Delta x in \text{ Diff}(\mathcal{B})}{\arg\max} \Big| \nabla_{\Delta x} J\left(\Delta x\right) \Big| \in \{\Delta x_{\min}, \Delta x_{\max}\} \tag{34}$$

10.5. Translation perturbation Δy

Figure 12. Samples for a y-translation perturbation, with $\boldsymbol{\kappa} = \Delta y \in \mathcal{B} = [0, 1]m$

Since there is no orientation change between both viewpoints, we retrieve a simple shear along the camera x-axis: once again an affine transform.

$$\boldsymbol{H}^{-1}(\Delta y) = \begin{bmatrix} 1 & -\frac{\Delta y}{z} & \frac{\Delta y y_c}{z} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{with } z \neq 0 \text{ (i.e. camera not on ground plane)} \tag{35}$$

$$\Big(u_0(\Delta y), v_0(\Delta y)\Big) = \left(u + \Delta y \left(\frac{y_c}{f} - \frac{v}{z}\right), \quad v\right) \tag{36}$$

$$\nabla_{\Delta y}^{\top} T^{-1}(\Delta y) = \left(\frac{y_c}{f} - \frac{v}{z}, \quad 0\right) \tag{37}$$

$$\text{Diff}(\mathcal{B}) = \mathcal{B} \tag{38}$$

Any $\Delta y \in \mathcal{B}$ is suitable to evaluate the Lipschitz constant $L = \sup_{\Delta y \in \text{Diff}(\mathcal{B})} \Big| \nabla_{\Delta y} J\left(\Delta y\right) \Big|$.

10.6. Translation perturbation Δz

Figure 13. Samples for a z-translation perturbation, with $\boldsymbol{\kappa} = \Delta z \in \mathcal{B} = [0, 1]m$

The simplified form results in a scaling along the camera y-axis.

$$\boldsymbol{H}^{-1}(\Delta z) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{z}{\Delta z + z} & \frac{\Delta z y_c}{\Delta z + z} \\ 0 & 0 & 1 \end{bmatrix} \tag{39}$$

$$\Big(u_0(\Delta z), v_0(\Delta z)\Big) = \Big(u, \quad \frac{zv + \Delta z y_c}{z + \Delta z}\Big) \tag{40}$$

$$\nabla_{\Delta z}^{\top} T^{-1}(\Delta z) = \Big(0, \quad -\frac{z(v - y_c)}{(z + \Delta z)^2}\Big) \tag{41}$$

$$\mathrm{Diff}(\mathcal{B}) = \mathcal{B} \setminus \{-z\} \qquad \text{(i.e. camera not on ground plane)} \tag{42}$$

$$\underset{\Delta z \in \mathrm{Diff}(\mathcal{B})}{\arg\max} \Big|\nabla_{\Delta z} J\left(\Delta z\right)\Big| \in \{\Delta z_{\min}, \Delta z_{\max}\} \tag{43}$$

11. Full closed-form expressions

We detail here the remaining closed-form expressions needed in Eq. (7). Note that the roll, pitch, yaw angles are not defined in the camera frame (C) (x-right, y-down, z-forward), but in a body frame (B) with x-forward, y-right, z-down. The transform $\boldsymbol{T}_{BODY}^{CAM}$ is thus needed to obtain the correct transforms. There is no translation between both frames:

$$\boldsymbol{T}_{BODY}^{CAM} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \left(\boldsymbol{T}_{CAM}^{BODY}\right)^{\top} \tag{44}$$

The transform from first to second viewpoint is given by $\boldsymbol{T}_{C_0}^{C} = [\boldsymbol{R}_{C_0}^{C} | \boldsymbol{t}_{C \to C_0}^{C}]$ with:

$$\boldsymbol{R}_{C_0}^{C} = \boldsymbol{R}_{BODY}^{CAM} \cdot \boldsymbol{R}_{B_0}^{B} \cdot \boldsymbol{R}_{CAM}^{BODY} = \boldsymbol{R}_{BODY}^{CAM} \cdot \left(\boldsymbol{R}_{B}^{B_0}\right)^{\top} \cdot \boldsymbol{R}_{CAM}^{BODY} \tag{45}$$

$$\boldsymbol{R}_{B}^{B_0} = \begin{bmatrix} \cos(\Delta\psi) & -\sin(\Delta\psi) & 0 \\ \sin(\Delta\psi) & \cos(\Delta\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\Delta\theta) & 0 & \sin(\Delta\theta) \\ 0 & 1 & 0 \\ -\sin(\Delta\theta) & 0 & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Delta\phi) & -\sin(\Delta\phi) \\ 0 & \sin(\Delta\phi) & \cos(\Delta\phi) \end{bmatrix} \tag{46}$$

$$\boldsymbol{t}_{C \to C_0}^{C} = \boldsymbol{T}_{BODY}^{CAM} \boldsymbol{t}_{B \to B_0}^{B} \qquad \text{with } \boldsymbol{t}_{B \to B_0}^{B} = [\Delta x, \Delta y, \Delta z]^{\top} \tag{47}$$

Similarly, the transform from ground plane to first viewpoint is given by $\boldsymbol{T}_{C_0}^{W} = [\boldsymbol{R}_{C_0}^{W} | \boldsymbol{t}_{W \to C_0}^{W}]$ with:

$$\boldsymbol{R}_{C_0}^{W} = \boldsymbol{R}_{B_0}^{W} \cdot \boldsymbol{W}_{CAM}^{BODY} \tag{48}$$

$$\boldsymbol{R}_{B_0}^{W} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{49}$$

12. Example: yaw deviation $\Delta\psi$

For the yaw deviation example, we set the pose of the first camera (i.e. $\phi, \theta, \psi, x, y, z$) to be arbitrary, and set $\Delta\phi = \Delta\theta = \Delta x = \Delta y = \Delta z = 0$ so that only the yaw deviation $\Delta\psi$ explains the motion from first to second viewpoint. In this context, $\boldsymbol{t} = \boldsymbol{0}$ and Eq. (7) reduces to the special case $\boldsymbol{H}(\Delta\psi) = \boldsymbol{K} \cdot \boldsymbol{R} \cdot \boldsymbol{K}^{-1}$, valid even if scene points are not co-planar (see [21, Sec. 8.4.5]). The yaw rotation then corresponds to a rotation about the camera y-axis:

$$\boldsymbol{H}(\Delta\psi) = \boldsymbol{K} \cdot \boldsymbol{R}_{\boldsymbol{C_0}}^{\boldsymbol{C}} \cdot \boldsymbol{K}^{-1} = \boldsymbol{K} \cdot \boldsymbol{R}_y(-\Delta\psi) \cdot \boldsymbol{K}^{-1} \tag{50}$$

As we need the inverse mapping, we can compute it directly as:

$$\boldsymbol{H}^{-1}(\Delta\psi) = \boldsymbol{K}\cdot\boldsymbol{R}_y^{-1}(-\Delta\psi)\cdot\boldsymbol{K}^{-1} \tag{51}$$

$$= \boldsymbol{K}\cdot\boldsymbol{R}_y(\Delta\psi)\cdot\boldsymbol{K}^{-1} \tag{52}$$

$$= \begin{bmatrix} f & s & x_c \\ 0 & f & y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\Delta\psi) & 0 & \sin(\Delta\psi) \\ 0 & 1 & 0 \\ -\sin(\Delta\psi) & 0 & \cos(\Delta\psi) \end{bmatrix} \begin{bmatrix} \frac{1}{f} & 0 & -\frac{x_c}{f} \\ 0 & \frac{1}{f} & -\frac{y_c}{f} \\ 0 & 0 & 1 \end{bmatrix} \tag{53}$$

$$= \begin{bmatrix} \cos(\Delta\psi) - \frac{x_c\sin(\Delta\psi)}{f} & 0 & \frac{\left(f^2+x_c^2\right)\sin(\Delta\psi)}{f} \\ -\frac{y_c\sin(\Delta\psi)}{f} & 1 & \frac{y_c(f(\cos(\Delta\psi)-1)+x_c\sin(\Delta\psi))}{f} \\ -\frac{\sin(\Delta\psi)}{f} & 0 & \cos(\Delta\psi) + \frac{x_c\sin(\Delta\psi)}{f} \end{bmatrix} \tag{54}$$

13. Maximization of $|\nabla_\kappa J|$

In [7], an upper bound on the Lipschitz constant is estimated as $\sup_{\boldsymbol{\kappa}\in\mathrm{Diff}(\mathcal{B})}\left|\nabla_{\boldsymbol{\kappa}}^\top J(\boldsymbol{\kappa})\cdot e_m\right|$ for coordinate m. We justify here the maximization of the independent components for the case where there is only one transform parameter ($m=1$):

$$|\nabla_{\boldsymbol{\kappa}} J(\boldsymbol{\kappa})| \triangleq |\nabla_{\boldsymbol{\kappa}}\left[\mathrm{LB}(\boldsymbol{\kappa}) - G(\boldsymbol{\kappa})\right]| \tag{55}$$

$$\triangleq \left|\nabla_{\boldsymbol{\kappa}}\left[\max_{j\in[\![1,q]\!]}\left\{\underline{\boldsymbol{w}}_j^\top\boldsymbol{\kappa}+\underline{\boldsymbol{b}}_j\right\} - G(\boldsymbol{\kappa})\right]\right| \tag{56}$$

$$\leq \left|\nabla_{\boldsymbol{\kappa}}\max_{j\in[\![1,q]\!]}\left\{\underline{\boldsymbol{w}}_j^\top\boldsymbol{\kappa}+\underline{\boldsymbol{b}}_j\right\}\right| + \left|\nabla_{\boldsymbol{\kappa}} G(\boldsymbol{\kappa})\right| \qquad \text{by triangle inequality of } |\cdot| \tag{57}$$

$$\triangleq |\underline{\boldsymbol{w}}^*| + \left|\nabla_{\boldsymbol{\kappa}}\left(I\circ T^{-1}\right)(\boldsymbol{\kappa})\right| \qquad \text{with } \underline{\boldsymbol{w}}^* = \max_{j\in[\![1,q]\!]}\left|\underline{\boldsymbol{w}}_j\right| \tag{58}$$

$$\triangleq |\underline{\boldsymbol{w}}^*| + \left|\nabla_{\boldsymbol{\kappa}} I\Big(u_0(\boldsymbol{\kappa}), v_0(\boldsymbol{\kappa})\Big)\right| \tag{59}$$

$$= |\underline{\boldsymbol{w}}^*| + \left|\frac{\partial I}{\partial u}\cdot\frac{\partial u_0}{\partial\boldsymbol{\kappa}}(\boldsymbol{\kappa}) + \frac{\partial I}{\partial v}\cdot\frac{\partial v_0}{\partial\boldsymbol{\kappa}}(\boldsymbol{\kappa})\right| \qquad \text{by chain rule} \tag{60}$$

$$\leq |\underline{\boldsymbol{w}}^*| + \left|\frac{\partial I}{\partial u}\cdot\frac{\partial u_0}{\partial\boldsymbol{\kappa}}(\boldsymbol{\kappa})\right| + \left|\frac{\partial I}{\partial v}\cdot\frac{\partial v_0}{\partial\boldsymbol{\kappa}}(\boldsymbol{\kappa})\right| \qquad \text{by triangle inequality of } |\cdot| \tag{61}$$

$$= |\underline{\boldsymbol{w}}^*| + \left|\frac{\partial I}{\partial u}\right|\cdot\left|\frac{\partial u_0}{\partial\boldsymbol{\kappa}}(\boldsymbol{\kappa})\right| + \left|\frac{\partial I}{\partial v}\right|\cdot\left|\frac{\partial v_0}{\partial\boldsymbol{\kappa}}(\boldsymbol{\kappa})\right| \tag{62}$$

In particular, a majorant can be chosen by maximizing each positive component independently:

$$\sup_{\boldsymbol{\kappa}\in\mathrm{Diff}(\mathcal{B})}|\nabla_{\boldsymbol{\kappa}} J(\boldsymbol{\kappa})| \leq |\underline{\boldsymbol{w}}^*| + \sup_{\boldsymbol{\kappa}\in\mathrm{Diff}(\mathcal{B})}\left|\frac{\partial I}{\partial u}\right|\cdot\sup_{\boldsymbol{\kappa}\in\mathrm{Diff}(\mathcal{B})}\left|\frac{\partial u_0}{\partial\boldsymbol{\kappa}}(\boldsymbol{\kappa})\right| + \sup_{\boldsymbol{\kappa}\in\mathrm{Diff}(\mathcal{B})}\left|\frac{\partial I}{\partial v}\right|\cdot\sup_{\boldsymbol{\kappa}\in\mathrm{Diff}(\mathcal{B})}\left|\frac{\partial v_0}{\partial\boldsymbol{\kappa}}(\boldsymbol{\kappa})\right| \tag{63}$$

14. Upper bound locations for gradient $|\nabla_{\Delta\psi} T^{-1}|$

In the yaw perturbation example, we follow Sec. 13 and derive a majorant for the gradient of the inverse transform $\left|\nabla_{\Delta\psi}^\top T^{-1}\right|$, more particularly along each coordinate by maximizing $\left|\frac{\partial u_0}{\partial\Delta\psi}\right|$ and $\left|\frac{\partial v_0}{\partial\Delta\psi}\right|$ independently.

u-coordinate. We study the variations of the gradient $\frac{\partial u_0}{\partial\Delta\psi}$ given by Eq. (12). We compute its derivative $\forall\Delta\psi\in\mathrm{Diff}(\mathcal{B})$ and study its sign:

$$\frac{\partial^2 u_0}{\partial\Delta\psi^2} = 2f\left[f^2+(u-x_c)^2\right]\frac{f\sin(\Delta\psi)+(u-x_c)\cos(\Delta\psi)}{\left[f\cos(\Delta\psi)-(u-x_c)\sin(\Delta\psi)\right]^3} \tag{64}$$

$$= \underbrace{\frac{2f\left[f^2+(u-x_c)^2\right]}{\left[f\cos(\Delta\psi)-(u-x_c)\sin(\Delta\psi)\right]^2}}_{>0}\left[\frac{f\sin(\Delta\psi)+(u-x_c)\cos(\Delta\psi)}{f\cos(\Delta\psi)-(u-x_c)\sin(\Delta\psi)}\right] \tag{65}$$

The derivative cancels out if $f\sin(\Delta\psi) + (u - x_c)\cos(\Delta\psi) = 0$, which yields:

$$\Delta\psi = \Delta\psi_0 \equiv \arctan\left(\frac{x_c - u}{f}\right) \pmod{\pi} \tag{66}$$

For our scope of robotics applications and cameras without wide fields of view, we limit ourselves to $\Delta\psi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, otherwise the plane of interest is not in the field of view.

In other cases where $\frac{\partial^2 u_0}{\partial \Delta\psi^2} \neq 0$, the gradient $\left|\frac{\partial u_0}{\partial \Delta\psi}\right|$ is monotonic, and its maximum occurs at the interval boundary $\psi_{\min}$ or $\psi_{\max}$. Overall, there are three candidate locations for the maximum of $\sup_{\Delta\psi \in \mathrm{Diff}(\mathcal{B})} \left|\frac{\partial u_0}{\partial \Delta\psi}\right|$: $\psi_{\min}$, $\psi_{\max}$, or $\Delta\psi_0$ if $\Delta\psi_0 \in [\psi_{\min}, \psi_{\max}]$.

v-coordinate. We study the variations of the gradient $\frac{\partial v_0}{\partial \Delta\psi}$ given by Eq. (12) and compute its derivative $\forall \Delta\psi \in \mathrm{Diff}(\mathcal{B})$:

$$\frac{\partial^2 v_0}{\partial \Delta\psi^2} = f(v - y_c) \cdot \frac{2\left[f\sin(\Delta\psi) + (u - x_c)\cos(\Delta\psi)\right]^2 + \left[f\cos(\Delta\psi) - (u - x_c)\sin(\Delta\psi)\right]^2}{\left[f\cos(\Delta\psi) - (u - x_c)\sin(\Delta\psi)\right]^3} \tag{67}$$

The derivative cancels out if:

$$2\left[f\sin(\Delta\psi) + (u - x_c)\cos(\Delta\psi)\right]^2 + \left[f\cos(\Delta\psi) - (u - x_c)\sin(\Delta\psi)\right]^2 = 0 \tag{68}$$

$$\Rightarrow \text{ (and) } \begin{cases} f\sin(\Delta\psi) + (u - x_c)\cos(\Delta\psi) = 0 \\ f\cos(\Delta\psi) - (u - x_c)\sin(\Delta\psi) = 0 \end{cases} \tag{69}$$

$$\Rightarrow \begin{cases} f\sin(\Delta\psi) = -(u - x_c)\cos(\Delta\psi) \\ f\cos(\Delta\psi) = (u - x_c)\sin(\Delta\psi) \end{cases} \tag{70}$$

$$\Rightarrow \begin{cases} \tan(\Delta\psi) = \frac{-(u - x_c)}{f} \\ \tan(\Delta\psi) = \frac{f}{u - x_c} \end{cases} \tag{71}$$

$$\Rightarrow \frac{-(u - x_c)}{f} = \frac{f}{u - x_c} \tag{72}$$

$$\Rightarrow f^2 = -(u - x_c)^2 \tag{73}$$

$$\Rightarrow u = x_c \text{ and } f = 0 \text{ (impossible as } f > 0) \tag{74}$$

Since $\forall \Delta\psi \in \mathrm{Diff}(\mathcal{B})$, $\frac{\partial^2 v_0}{\partial \Delta\psi^2} \neq 0$, the gradient $\left|\frac{\partial v_0}{\partial \Delta\psi}\right|$ is monotonic, and its maximum occurs at the interval boundary $\psi_{\min}$ or $\psi_{\max}$.

15. Derivation of f_{bound}

By considering the subdomain midpoint, Equation 18 in Batten et al. [7] proposes to use $f_{\text{bound}} = f\left(\frac{\kappa_1 + \kappa_2}{2}\right) + \frac{L}{2}(\kappa_2 - \kappa_1)$. In our implementation, we propose a slightly different bound by working rather with the subdomain extremities. Let $\kappa \in [\kappa_1, \kappa_2] \subset \mathbb{R}$. Using the Lipschitz property of f, we derive the following bound for $f(\kappa)$:

$$\begin{aligned} &\begin{cases} f(\kappa) \leq f(\kappa_1) + L \cdot |\kappa_1 - \kappa| = f(\kappa_1) + L \cdot (\kappa - \kappa_1) \\ f(\kappa) \leq f(\kappa_2) + L \cdot |\kappa_2 - \kappa| = f(\kappa_2) + L \cdot (\kappa_2 - \kappa) \end{cases} \\ &\Rightarrow 2 \cdot f(\kappa) \leq f(\kappa_1) + f(\kappa_2) + L \cdot (\kappa_2 - \kappa_1) \qquad \text{summing up both lines} \\ &\Rightarrow f(\kappa) \leq \frac{1}{2} \cdot \Big[f(\kappa_1) + f(\kappa_2) + L \cdot (\kappa_2 - \kappa_1)\Big] \triangleq f_{\text{bound}} \end{aligned} \tag{75}$$

16. Models

To verify robustness against the VNN datasets [9], we use the official benchmark networks available from the competition repositories at https://github.com/VNN-COMP (see Tab. 5). There is no existing VNN benchmark for runway visibility, so we train a custom runway classifier on the LARD dataset [14], network and training details are given below.

Dataset	Model

MNIST	`mnist-net_256x2.onnx`
	`mnist-net_256x4.onnx`
	`mnist-net_256x6.onnx`

CIFAR-10	`cifar_base_kw.onnx`
	`cifar_wide_kw.onnx`
	`cifar_deep_kw.onnx`

GTSRB	`3_30_30_QConv_16_3_QConv_32_2_Dense_43_ep_30.onnx`
LARD	Custom (based on `resnet_4b.onnx`)

Table 5. Detail of models used in our experiments.

Runway classifier details. We only use data for which along-track distance to the runway is available (e.g. `LARD_train.csv` and `LARD_test_synth.csv`). We stratify the data per approach, and keep 80% of approaches for training and validation, and 20% for testing. In order to make models compatible with the current scalability of formal verification techniques, we scale all images down to 32×32. We add new labels for classification instead of object detection: positive instances are images with a runway occupying more than 25 pixels, negative instances are images with a runway smaller than 1 pixel. Runway sizes in-between are ignored for better class separation. The model is a small ResNet architecture `resnet_4b.onnx` introduced in the 2021 VNN competition. We train for 100 epochs with an Adam optimizer and an initial learning rate of 3×10^{-4}. A scheduler reduces this rate by a factor of 0.5 every 20 epochs without loss improvement. We do not use any robustification techniques during training, but we use data augmentation (horizontal flip, random rotation up to $10°$, random translation between 0.1 and 0.2, and random perspective up to 0.2 distortion). Despite the dataset imbalance, we use a simple weighted cross-entropy loss function. This approach produces reasonable results with a 0.96 area under the ROC curve (161 true positives, 36 false positives, 433 true negatives, 26 false negatives). The verification instances are then sampled from the true positives and the true negatives.

17. Extended results

To complete Tab. 2, this section presents extended robustness results on additional networks of the VNN-COMP benchmarks. We indicate the number of timeouts in brackets, if any. A high number of timeouts indicates a potential need to increase the timeout limit and leave more time for the verifier to output a certified result.

Perturbation	Range $\mathcal{B}$	MNIST			CIFAR-10			GTSRB
		`x2`	`x4`	`x6`	`base`	`wide`	`deep`	`30_30`
$\boldsymbol{\Delta\phi}$ (roll)	$[\mathbf{0}, \mathbf{5}]^\circ$	61	63	48	69	73 (11)	57 (4)	0
$\boldsymbol{\Delta\theta}$ (pitch)	$[\mathbf{0}, \mathbf{5}]^\circ$	5	4	2	10 (1)	5 (47)	2 (4)	0
$\boldsymbol{\Delta\psi}$ (yaw)	$[\mathbf{0}, \mathbf{5}]^\circ$	25	15	7 (3)	7 (1)	6 (50)	1 (4)	0
$\boldsymbol{\Delta x}$	$[\mathbf{0}, \mathbf{1}]\ \boldsymbol{m}$	50	52	40	21	19 (33)	9 (13)	0
$\boldsymbol{\Delta y}$	$[\mathbf{0}, \mathbf{1}]\ \boldsymbol{m}$	73	74	62	83	85 (3)	71 (4)	0
$\boldsymbol{\Delta z}$	$[\mathbf{0}, \mathbf{1}]\ \boldsymbol{m}$	53	53	42	24	33 (28)	13 (18)	0

Table 6. **Ratio (%) of robust cases to non-affine perturbations.**