# 

# **No GPS no problem! Democratising aerial navigation via robust and data-scalable computer vision.**

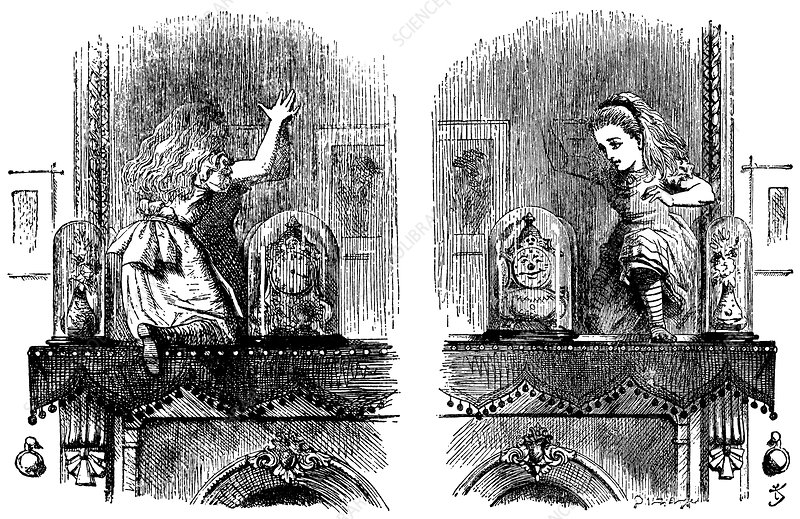
*Do you believe like us in pushing CV advances in the real world? Will you be a champion in proving CV advances benefits to the world of aviation? Then join us and earn:*

**→ Respect and fame:** As you, your team product or company being the leader in the real-world challenge of aerial navigation. Leaderboard open for all with no restrictions.

→ **The open data champions prize:** **10 000 CHF** (8 000 CHF for 1st place and 2 000 CHF for second) for the champions on our leaderboard that choose to share to the world (open source) their solution (git and model explanation report) for the benefit of all.

The value of research shines best when it touches everyday life. Today's scientific journals in Computer Vision (CV) provide exciting developments that promise a future in which CV systems can automise navigation, control of ground and aerial vehicles. Thus a lasting impact on our daily lives beyond the annals of academic journals. Traditionally, however, academic advancements are focusing on developing theoretical solutions optimised to perform over a given data domain. This being a set of camera images collected at specific geographic location, trajectory path, camera type, weather, season, date, etc. Those algorithms can rarely be proven to generalise to real-world large-scale applications. Those would require an economically prohibitive amount of training data describing all possible situations and locations in time and space. This poses a challenge to democratisation and realisation of the autonomous/GPS-free promise. As rising countermeasure, synthetic data and rendering engines have attempted to fill in the data hunger of the CV models ([TOPO-DataGen](https://github.com/TOPO-EPFL/TOPO-DataGen) open source example for air nav domain). Two paths exist in bridging the gap between the real scarce data and abundant synthetic:

1. Either making the synthetic data highly representative of reality or
2. Making models robust to sim-to-real traverse/gap



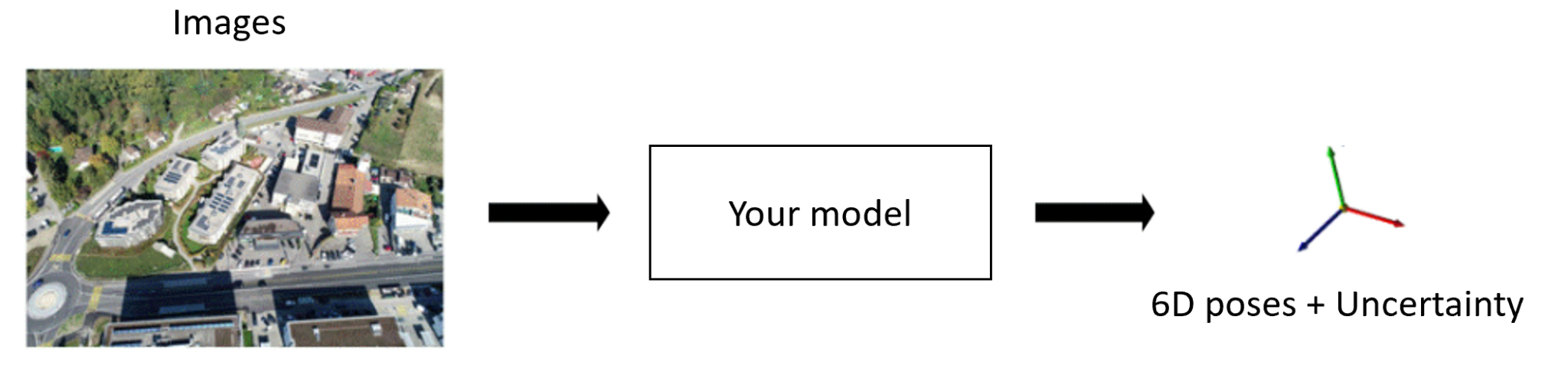
*Through the Looking-Glass, Alice Pushes Through the Mirror (between real and synthetic data)*

We believe that the real-challenge of today's CV is to develop and validate a design philosophy for algorithms that can work **robustly** and in an **economically/data scalable way.** In other words:

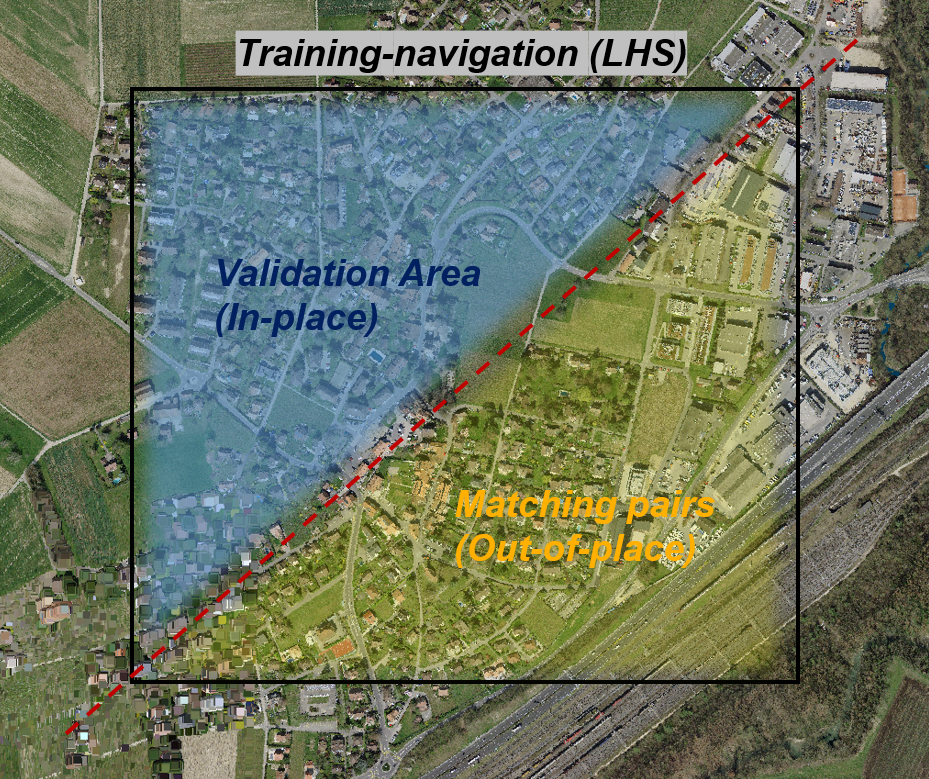
* Being able to train with limited or even no real/domain specific data (**zero-shot**). Those leveraging synthetic data generated by open source tools such as [TOPO-DataGen](https://github.com/TOPO-EPFL/TOPO-DataGen).
* Capable of producing accurate output with trustworthy **uncertainty statements** that can capture accurately the prediction errors. [4]

**The challenge, should you choose to accept it.**

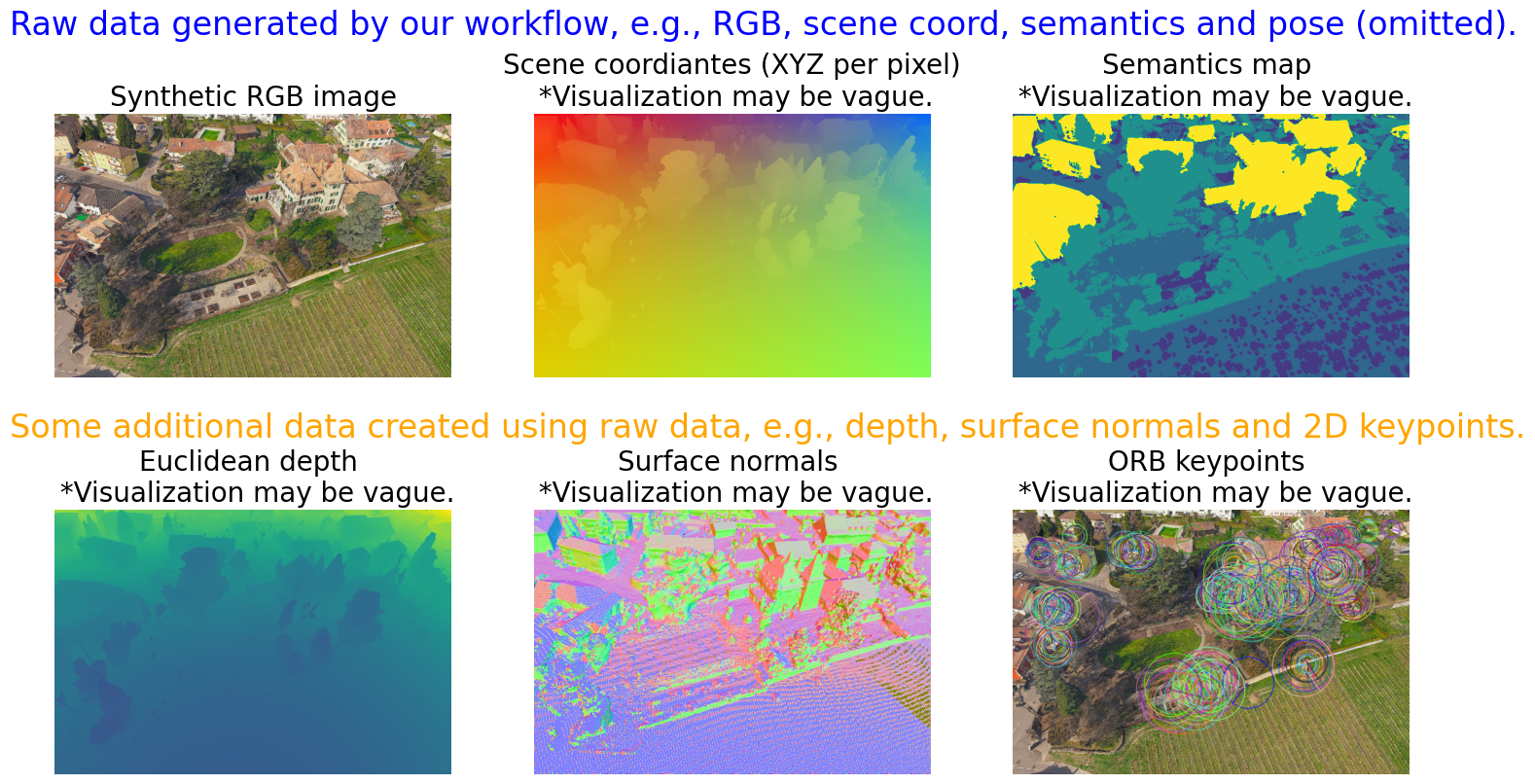
Today aerial autonomous systems for navigation and control are heavily dependent on the robustness of GNSS reception. Lack of thereof (terrain, weather, or adversarial spoofing) can lead to loss of autonomous system absolute orientation. This in the medium to long run (error will always accumulate in time with dead reconing) can be unsafe for operation, especially on beyond line of sight missions. Despite significant progress in Computer Vision, most learning-based approaches target at a single domain and require a dense database of geo-tagged images to function well. Or at least a calibration set of real images taken closely from the domain of [1,2] operation. Several industry attempts are made in the same direction, however, understandably without an official benchmark or validation.

We challenge you to prove your approach can work **robustly** and in an **economically/data scalable way**. To do this by proving you can compute accurate 6D camera poses with known uncertainty on our challenge validation dataset. To do this by having only access to:

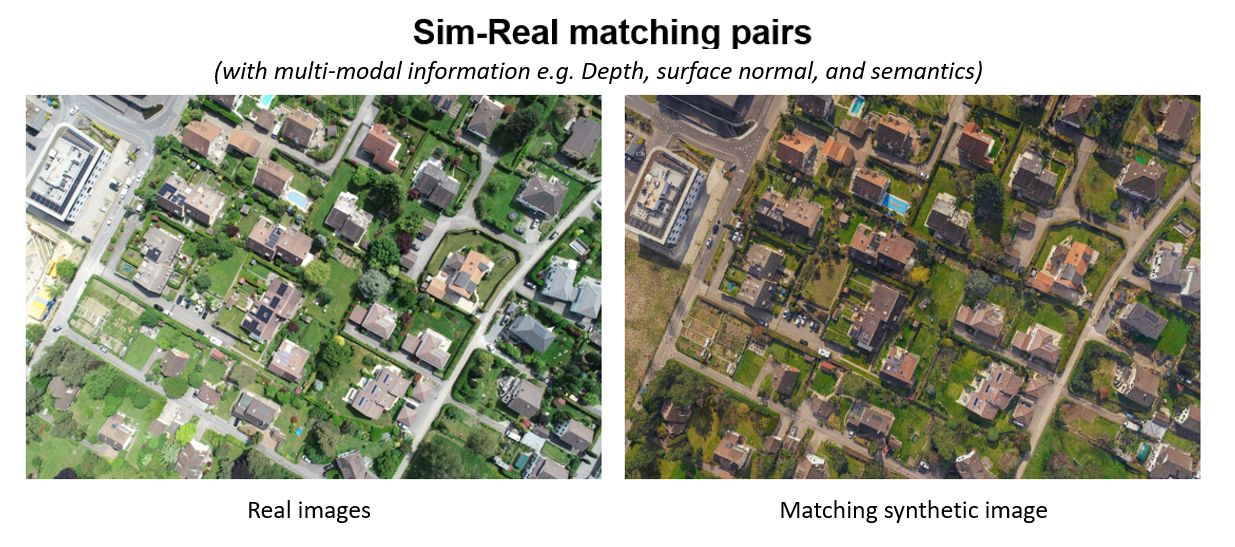
# **The Datasets**



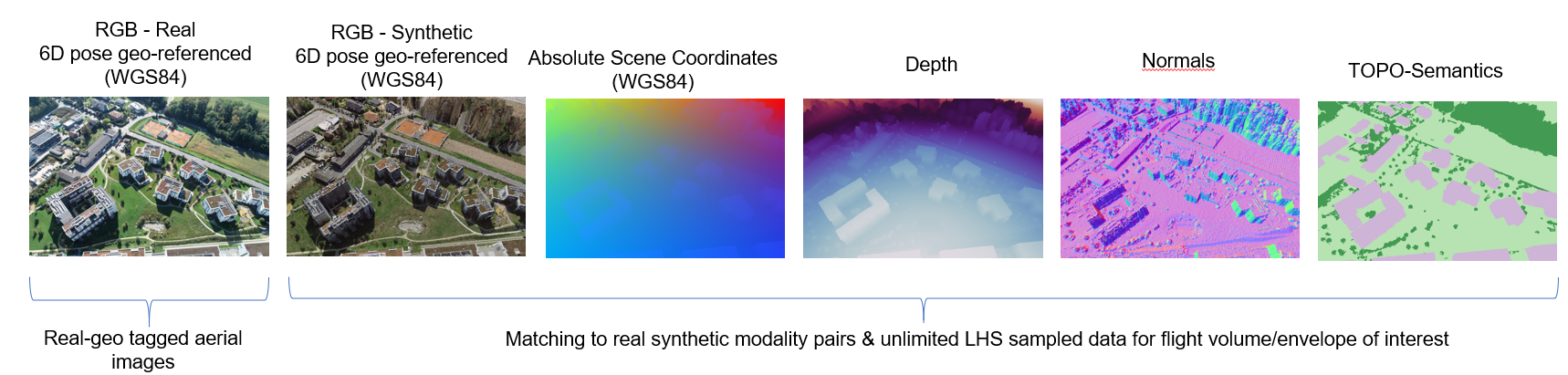
1. **Training–navigation (In-place LHS)\*** A large dataset consisting of synthetic rendered images with ground truth poses, matching scene coordinates, depth, normals, semantics for the navigation area of interest. You can observe an example below:



1. **Matching pairs (Out-of-place)** A subset of synthetic and real image pairs taken from similar-looking places but of geographically different locations as an aid with the sim-to-real transfer and uncertainty estimation calibration.



All of the matching pairs have as well related ground-truth matching modalities generated by [TOPO-DataGen.](https://github.com/TOPO-EPFL/TOPO-DataGen)



1. **Validation Area (In-place)** A validation dataset consisting of real images on which you will be marked on our automated leaderboard against all fellow participants.

\*Notation explanation:

* What is “LHS”?
  + LHS stands for the [Latin Hypercube Sampling](https://en.wikipedia.org/wiki/Latin_hypercube_sampling), a statistical method for efficiently sampling values from a multidimensional distribution. We use the LHS method to generate the synthetic data viewpoints (i.e., the 6D camera poses) within the area of interest. This ensures that the generated data points are randomly distributed and span over the benchmark region densely enough.
* What is the purpose of having “In-place” and “Out-of-place” datasets?
  + Intuitively, the "In-place" area is where the final localization validation is carried out. To truly highlight the model's robustness and generalisation capability, the real images of In-place areas are with hidden meta data as metric datasets. We only provide **synthetic data** of the In-place site for model training.
  + We assume that the model can access some **real & synthetic data** from the neighbouring areas, dubbed “Out-of-place.” This way, you can train the model to generalise from sim to real and **avoid overfitting** the In-place real data.

## Links

The data splits and link to the repositories:

| **Data set name** | **Type** | **Number of images** | **Link** |
| --- | --- | --- | --- |
| Training-navigation (In-place LHS) | zero-shot | 15000 | [train.zip](https://zenodo.org/record/6650721/files/train.zip?download=1) |
| Matching pairs (Out-of-place) | Sim-to-Real | 1197 |
| Validation (In-place) | Validation | 300 | [validation.zip](https://zenodo.org/record/6669020/files/validation.zip?download=1) |

## Coordinates system (position)

In order to facilitate the dataset processing, all coordinates (expressed in ECEF coordinates system) of the provided datasets have been reduced by subtracting all from a reference amount (ECEF column 1 bellow - “local reference”). In the second column an example with current image position is shown in ECEF with the corresponding “relative/reduced” coordinate with respect to local reference. In the final column one can see the corresponding WGS 84 coordinates for the “current position”. All final submission of results must be provided in WGS 84 format for final marking.

|  | ECEF  [Epsg:4978](https://epsg.io/4978) **local reference** | ECEF  [Epsg:4978](https://epsg.io/4978) **current position** | ECEF  [Epsg:4978](https://epsg.io/4978) **relative/reduced** position W.R. to local ref.  as provided in dataset/meta data |  | WGS 84  **current position**  [epsg:4979](https://epsg.io/4979)  [Latitude/Longitude/Height <--> ECEF via J-Script (nps.edu)](https://www.oc.nps.edu/oc2902w/coord/llhxyz.htm) |
| --- | --- | --- | --- | --- | --- |
| X [m] | 4366983.403482145 | 4366993.403482145 | 10 | Latitude [deg] N | 46.53561 |
| Y [m] | 500859.6799224778 | 500879.6799224778 | 20 | Longitude [deg] E | 6.54304 |
| Z [m] | 4606715.270014311 | 4606745.270014311 | 30 | Height [m] | 470.2 |

## Coordinates system (orientation)

Bellow you can see an example of the orientation reference coordinate system. North being the reference for Yaw and horizontal for Pitch and Roll.



# Submission

## Format

To submit your results to the leaderboard you must construct a submission zip file containing a CSV file that must have the following columns (without column names):

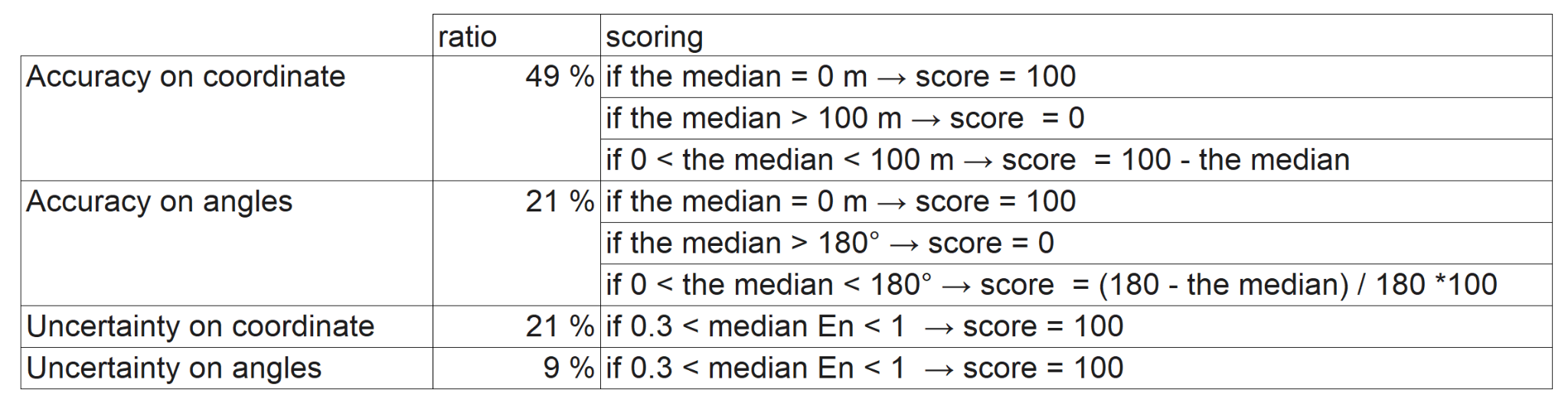
* Column 1 : estimate X coordinate (longitude) [deg]
* Column 2 : estimate Y coordinate (latitude) [deg]
* Column 3 : estimate Altitude [metres]
* Column 4 : estimate Azimuth [deg]
* Column 5 : estimate Tilt [deg]
* Column 6 : estimate Roll [deg]
* Column 7 : uncertainty X coordinate (longitude) [deg],
* Column 8 : uncertainty Y coordinate (latitude) [deg],
* Column 9 : uncertainty Altitude [metres],
* Column 10 : uncertainty Azimuth [deg]
* Column 11 : uncertainty Tilt [deg]
* Column 12 : uncertainty Roll [deg]

Notes:

1. Uncertainty: (1 sigma - standard Uncertainty [3])
2. Position: Coordinates are in WGS84 coordinate system and height in [metres].

## Evaluation Criterias

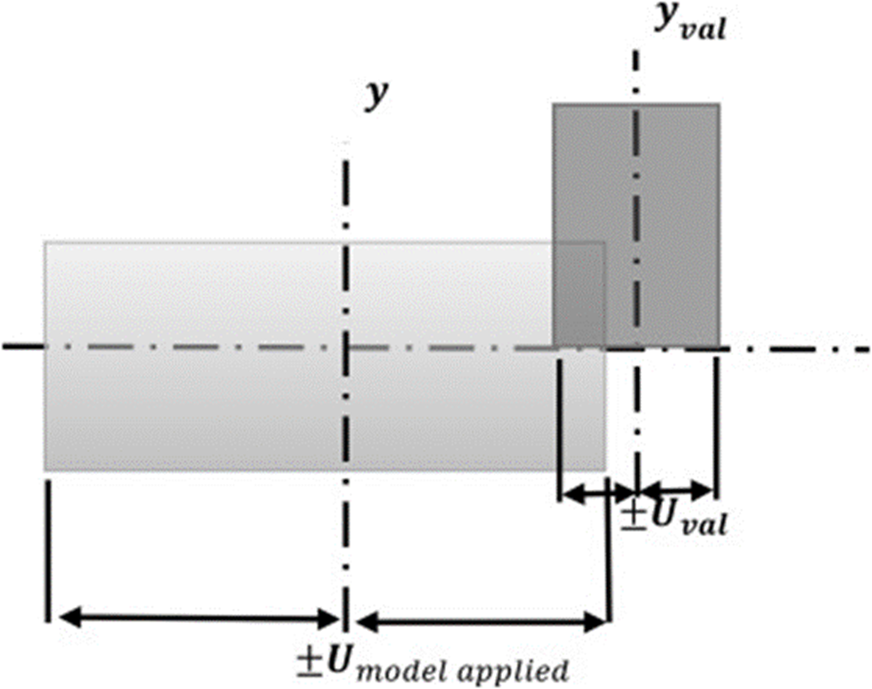
Accuracy on positioning (70 %), Providing a meaningful uncertainty statement (30 %). If Uncertainty statement is provided, competitor will be judged on the median of their lowest uncertainty 5 % image. We do this to motivate accurate uncertainty estimates. 5 % of estimates is considered acceptable as for absolute updates are required in only in relatively low frequency (conisdering good dead reconing scheme is present).

****

**Error calculation metric - Mean Average Error (MAE)**

****

**Uncertainty quality metric - En [4]**

****

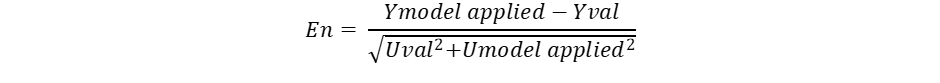
An Uncertain statement is useful if:

a) It can help one filter systematically images below a certain error threshold

b) Capture within it the actual magnitude of the errors

Criteria for uncertainty statement? An evaluation parameter inspired by the coordinate metrology standards - [4]

Where for each single estimation of each pose En value should be estimated



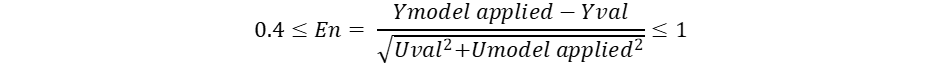
Ymodel applied = Prediction (Neural Network)

Yval = Reference (ground truth)

Uval = Uncertainty of reference (RTK GNSS fusion) = 2.2 cm – 1 sigma and 1.1 deg - 1 sigma

Umodel applied = Uncertainty of Neural Network – 1sigma

Once En value is calculated for each pose/uncertainty pairs a mean of those values is taken for each DoF. The final En value is then calculated. It can be used to judge the overal pose uncertainty quality. If En is in the provided range as shown below it will be considered useful (points gained). The provided range is given as such to ensure U captures errors without unecesearly enlarging the Uncertainty bounds.



# **Terms and Conditions**

## Data

* The datasets and competition should be cited acordingly in any further use or mention.
* The data should not be used for any other purpouces than academic or industrial research.

## Judging the entries

* Each participant must create a CodaLab account to register. Only one account per user is allowed.
* Evaluation criteria will be used to define the overal winner. Best results = winners. However only participants who adhore to the bottom mentioned conditions will be considered eligible for receiving the open source monetary prizes 1st and 2nd place.
* To receive the monterey prize the top runners must:
  + Deliver to the Competition Organizer the final model’s software code (training and inference code) as used to generate the winning submission and associated documentation written in English. The delivered software code must be capable of generating the winning submission and contain a description of resources required to build and/or run the executable code successfully. Their code will be tested internaly on hidden for the participants validation dataset.
  + To receive the monterey prize the top runners must also deliver the software code well documented.
  + Must agree for their git and report to be submitted open source for the community to benefit.

# Prizes attribution

If a team wins a monetary prize, the Competition Organizer will allocate the prize money in even shares between team members unless the team unanimously contacts the Competition Organizer within three business days following the submission deadline to request an alternative prize distribution.

# **Schedule**

* Starts: June XX, 2022, 8 a.m.
* Ends : September XX, 2022, 8 a.m.

# **Organization committee**

Scientific supervisor : **Iordan Doytchinov** [Profile](https://people.epfl.ch/iordan.doytchinov?lang=en)

Technical manager : **RÃ©gis Longchamp** [Profile](https://people.epfl.ch/regis.longchamp)

# **Prizes**

* 1st place: 8,000 CHF
* 2th place: 2,000 CHF

See the terms and conditions for further details

# **References**

[1][Previous related project website](https://crossloc.github.io/) (coordinate scene regression based architecture)

[2][Competing State of the art architecture](https://psarlin.com/pixloc/) (feature regression based localisation)

[3][Standard Uncertainty](https://physics.nist.gov/cgi-bin/cuu/Info/Constants/definitions.html#:~:text=Standard%20Uncertainty%20and%20Relative%20Standard%20Uncertainty%20Definitions,is%20not%20equal%20to%200)

[4][How can you judge the quality of your Uncertainty?](https://www.iso.org/obp/ui/#iso:std:iso:ts:15530:-4:ed-1:v1:en)