PFE Projects Book

ReDX

Email us at contact@redxt.com with your CV and a letter of motivation about the project you are interested in and we will get back to you.

Academic Year 2025/2026

Project 8: CFD-based Multi-Drone Vertiport Downwash & Urban Interaction Study

Description

This 6-month project builds an end-to-end workflow to evaluate how multirotor drones interact with each other and with the built environment during takeoff, landing, and hover, for the purpose of vertiport design. The work starts from real drone CAD, establishes a repeatable geometry cleaning pipeline in Blender, generates meshes suitable for rotorcraft flow, then runs OpenFOAM simulations for:

1. single-drone baseline hover/near-ground,
2. two/three-drone proximity operations,
3. building/structure/ground effects, and
4. weather sensitivity (wind, gusts, turbulence intensity).

The output is a simulation toolkit + a set of design-relevant insights (safety distances, layout guidance, operational envelopes).

PFE project Goals:

1. Create a library of drone geometries (at least 3/4 different multirotor CAD models).
2. Standardize geometry cleaning in Blender (or equilvalent).
3. Build a robust OpenFOAM meshing workflow (for examples: snappyHexMesh) for rotorcraft + nearby structures.
4. Establish a baseline single-drone hover simulation.
5. Simulate ground effect and landing/takeoff phases.
6. Study multi-drone interaction (2/3 drones): side-by-side, stacked altitude, crossing paths, approach/departure separation distances.
7. Add weather sensitivity: e.g. crosswind direction sweep via boundary conditions.

Required skills:

Must-have

- Linux basics, scripting (Python or Bash) for automation
- CAD basics
- OpenFOAM fundamentals: case setup, turbulence models (RANS/URANS), snappyHexMesh, post-processing in ParaView
- Solid understanding of CFD concepts

Nice-to-have:

- Rotor modeling knowledge: actuator disk, MRF, AMI, overset, or dynamic mesh approaches
- HPC usage

Deliverables:

1. **Drone geometry pack**
   - Cleaned CAD/STL/OBJ files + documentation of cleaning steps and conventions
2. **Reusable simulation templates**
   - OpenFOAM case templates for:
     - Single drone hover (baseline)
     - Ground effect sweep
     - Drone near building/wall/pad
     - Multi-drone (2/3) proximity scenarios
   - Meshing templates (snappyHexMesh dictionaries) and automation scripts
3. **Final report + handover**
   - A polished technical report (methods, verification/validation approach, limitations, results)
   - A reproducible repo structure (scripts + instructions) so the workflow can be extended to new drone models/sites

Duration and other details:

- **Recommended period:** 6 months (full-time PFE)
- **Suggested phased plan:**
  - **Month 1:** Collect CAD + cleaning pipeline + initial meshing template
  - **Month 2:** Single-drone baseline hover
  - **Month 3:** Ground effect + takeoff/landing represented as staged heights/ thrust settings
  - **Month 4:** Buildings/structures interaction (vertiport pad + nearby obstacles)
  - **Month 5:** Multi-drone interaction matrix (2–3 drones, spacing/altitude/path variants)
  - **Month 6:** Weather sensitivity + consolidate guidance + final report + packaging
- **Compensation:** Monthly stipend with potential end-of-internship performance bonus and potential paper publication co-authorship

Project 9: AI Surrogate Models for Near Real-Time Vertiport Downwash & Urban Interaction Predictions

Description

This 6-month project delivers an AI surrogate layer for a vertiport digital twin to support Advanced Air Mobility (AAM) planning and decision support. The final surrogate model learns from high-fidelity CFD simulation outputs (and selected external inputs such as wind profiles or turbulence intensity) to produce near real-time predictions of operationally relevant flow metrics (e.g., downwash footprint, peak velocities, recirculation zones, ground/building interaction effects, and safety envelopes).

The work spans the full ML lifecycle: building a clean and reproducible dataset pipeline, training/evaluating surrogate models, calibrating uncertainty/robustness, and packaging inference into a lightweight deployable component that integrates into the platform and aligns with the CFD team's validation workflow.

PFE project goals

1. **Define the surrogate problem** clearly (inputs outputs) aligned with vertiport decisions: spacing/layout, operational envelopes, weather sensitivity.
2. **Build a clean dataset pipeline** from CFD outputs + configuration metadata (drone geometry class, rotor model parameters, layout, wind BCs, turbulence settings).
3. **Create a standardized data format** (scientific-friendly, reproducible, versioned) with train/val/test splits that avoid leakage across scenarios.
4. **Train baseline surrogate models** for core tasks (e.g., downwash footprint / near-ground velocity field / "hazard map").
5. **Improve model performance** via architecture tuning, conditioning, and loss design (spatially weighted losses, physics-inspired regularization where useful).
6. **Assess generalization** to new layouts/weather/drone configurations (within defined bounds).
7. **Add uncertainty awareness** (e.g., ensembles or calibrated confidence) to support risk-aware planning.

What you will do

- Build a **clean dataset pipeline** from simulation outputs/inputs (parsing, alignment, normalization, metadata, versioning).
- Train and evaluate **surrogate models** for fast prediction of flow/hazard metrics.
- Package inference in a **lightweight deployable form**.

Required skills

Must-have

- Strong PyTorch skills and solid ML engineering practices (clean training loops, logging, checkpoints, reproducibility).
- Experience with regression/prediction on spatial and/or temporal data (images, grids, sequences, or fields).
- Comfort with scientific data formats and reproducible experiments (e.g., HDF5/NetCDF/VTK-like outputs, metadata handling).
- Good fundamentals in ML evaluation (generalization tests, ablations, error analysis).

Nice-to-have

- Surrogate / operator learning experience (e.g., neural operators) or GNNs for geometry-aware learning.
- Uncertainty quantification (ensembles, calibration).
- CUDA performance basics / inference optimization.
- Familiarity with CFD concepts (turbulence, boundary conditions, non-dimensionalization) to communicate efficiently with the CFD team.

Deliverables

1. **Dataset + data contract**
   - Versioned dataset structure (inputs/outputs/metadata)
   - Data loader utilities + documentation
2. **Trained surrogate models + benchmarks**
   - At least one strong baseline + one improved model
   - Evaluation report across scenario splits (layout/weather/drone configs)
   - Latency and memory footprint measurements for deployment targets
3. **Final report + handover**
   - Polished technical report: methodology, experiments, validation approach, limitations, safe use domain
   - Reproducible repo structure (training, evaluation, inference, configs) so the workflow extends to new sites/drone models

Duration and other details

- **Recommended period:** 6 months (full-time PFE)

Suggested phased plan (high-level):

- **Month 1:** Data contract + dataset pipeline v1 + baseline targets/metrics + initial train/val/test strategy
- **Month 2:** Baseline surrogate model(s) + first end-to-end results + error analysis
- **Month 3-4:** Model improvement (conditioning, losses, architecture) + robustness/generalization tests
- **Month 5:** Expand coverage (more scenarios) + uncertainty estimation approach + reliability analysis
- **Month 6:** Consolidate results + validation summary + final report + handover + publication-ready figures (if applicable)
- **Compensation:** Monthly stipend with potential end-of-internship performance bonus and potential paper publication co-authorship

ReDX

Project 10: Drone-Based Vehicle Detection, Deblurring, and Feature Extraction in Adverse Weather & Turbulence

Description

This 6-month project builds an end-to-end computer vision pipeline for a surveillance drone operating in harsh environments where imagery is frequently degraded by motion blur, atmospheric turbulence (sand/heat shimmer), rain, and low visibility. The system performs three coupled tasks in a reliable, real-time oriented workflow:

1. **Vehicle detection and tracking** from aerial video streams,
2. **Image restoration / deblurring** to recover readable and identifiable details, and
3. **Feature extraction** to surface actionable information such as license plate number, vehicle make/model, and text on vehicle body.

The output is a deployable prototype (or simulation-ready benchmark) that can run on edge hardware or a ground station, together with a rigorously constructed dataset and evaluation protocol that reflects real operating conditions.

PFE project goals

1. Build a benchmark dataset representative of aerial surveillance with controlled degradation types (sand/heat/rain + motion blur).
2. Establish a vehicle detection + tracking baseline that is robust to blur and turbulence.
3. Design and train a deblurring / restoration model optimized for downstream recognition (not just visual quality).
4. Train a license plate recognition model and a vehicle make/model classifier, plus text spotting for markings on the car.
5. Integrate all components into a single pipeline with clear interfaces and efficient runtime.
6. Quantify the benefit of deblurring on downstream tasks (plate accuracy, make/model accuracy, text read accuracy).
7. Implement failure detection / confidence scoring (when the system is unsure and should request closer pass / better view).
8. Package an inference demo with replayable videos and a simple UI overlay for detected objects + extracted attributes.

What you will do

- Build a data pipeline: ingest videos/images, annotate or adapt labels, generate synthetic degradations (sand/heat shimmer/rain blur).
- Train and evaluate detection + tracking models on clean and degraded imagery.
- Train a deblurring model (conditional restoration) and evaluate it using both image metrics and task metrics.

- Implement feature extraction modules: license plate detection/recognition, make/model classification, and text spotting.
- Build a real-time oriented inference pipeline with optimized batching and conditional execution.
- Work with the team to define deployment constraints and integrate outputs into the surveillance workflow.

Required skills

Must-have

- Strong PyTorch skills + clean ML engineering practices (reproducible experiments, logging, checkpoints).
- Experience with computer vision tasks.
- Familiarity with image restoration (deblurring/denoising/super-resolution) and loss design.
- Comfort with large-scale data handling and scientific reproducibility.

Deliverables

1. **Dataset + degradation benchmark**
    - Curated dataset splits (clean + degraded variants)
    - Degradation generator (sand turbulence / heat shimmer / rain + motion blur) with configurable parameters
    - Documentation of labeling schema and evaluation protocol
2. **Models and baselines**
    - Vehicle detector + tracker baseline with performance report
    - Deblurring/restoration model with ablations
    - Feature extraction suite:
        - License plate detection + recognition
        - Vehicle make/model classifier
        - Text spotting for vehicle markings
3. **Integrated pipeline demo**
    - End-to-end inference script (video in → overlay out)
    - UI overlay: bounding box + ID + confidence + extracted attributes (plate/make/text)
    - Runtime profiling (FPS/latency) and recommended deployment configuration
4. **Final report + handover**
    - Polished technical report: methods, experiments, metrics, limitations, risk cases
    - Reproducible repo (training/eval/inference/configs) + onboarding instructions

Duration and other details

- **Recommended period:** 6 months

Suggested phased plan (high-level):

- **Month 1:** Dataset definition + ingestion + annotation strategy + degradation pipeline
- **Month 2:** Vehicle detection + tracking baseline (clean vs degraded evaluation)
- **Month 3:** Deblurring model v1 + demonstrate uplift in plate readability/ recognition
- **Month 4:** Feature extraction models

- **Month 5:** End-to-end pipeline integration + confidence scoring + runtime optimization
- **Month 6:** Robustness testing, edge-readiness packaging, final report + handover
- **Compensation:** Monthly stipend with potential end-of-internship performance bonus and potential paper publication co-authorship