Handling Pitch Variations for Visual Perception in MAVs: Synthetic Augmentation and State Fusion

E. Cereda[1],* D. Palossi[1,2], A. Giusti[1]
[1]Dalle Molle Institute for Artificial Intelligence (IDSIA) – USI and SUPSI – Lugano, Switzerland
[2]Integrated Systems Laboratory (IIS) – ETH – Zürich, Switzerland

ABSTRACT

Variations in the pitch of a Micro Aerial Vehicle affect the geometry of the images acquired by its on-board cameras. We propose and evaluate two orthogonal approaches to handle this source of variability, in the context of visual perception using Convolutional Neural Networks. The first is a training data augmentation method that generates synthetic images simulating a different pitch than the one at which the original training image was acquired; the second is a neural network architecture that takes the drone's estimated pitch as an auxiliary input. Real-robot quantitative experiments tackle the task of visually estimating the pose of a human from a nearby nano-quadrotor; in this context, the two proposed approaches yield significant performance improvements, up to +0.15 in the R^2 regression score when applied together.

1 INTRODUCTION

In autonomous MAVs, convolutional neural networks (CNNs) are frequently adopted to process on-board camera images to solve perception problems [1, 2, 3, 4]. This is especially true in nano-sized MAVs (i.e., sub-10cm span and few tens of grams weight), where tight computational and memory resources highly constrain the affordable methods.

To achieve consistent performance, these CNNs must be robust to sources of variability that may occur during deployment. One such example is the continuously-changing orientation of the camera, caused in quadrotors by pitch and roll changes required to generate accelerations; while larger drones feature mechanically, optically or electronically stabilized cameras, this is not the case for nano-quadrotors.

In this work, we consider the task of human pose estimation, leveraging the PULP-Frontnet CNN [4] to visually estimate the relative pose of a person from a low-resolution image acquired by a nano-quadrotor flying in its proximity. To ensure good performance in the field, this model should be as insensitive as possible to changes in drone attitude.

As a **main contribution**, we propose an ad-hoc training data augmentation method that synthesizes images as if acquired from a different pitch than the one at which the original image was acquired; this improves robustness to drone pitch changes with no impact on inference time; the approach is therefore well suited for deployment on nano-quadrotors.

As a **secondary contribution**, we propose to provide the camera pitch angle as an auxiliary input to the CNN, to facilitate its perception task. In fact, a good pitch estimate is always available from the quadrotor's inertial measurement unit (IMU) and state estimation subsystem; because pitch affects the observed image geometry, we hypothesize that its knowledge helps the model to correctly interpret the image data. Similarly, neuroscience research [5, 6] found that information from the vestibular system, which encodes the orientation of the head, contributes to the human visual perception.

After reviewing related work (Section 2), our contributions are described in Section 3. In Section 4 we describe our experimental setup; in Section 5, quantitative experiments demonstrate significant benefits of both our proposed contributions.

2 RELATED WORK

Camera images, like most high-dimensional data, often allow for large-scale variations which nonetheless have no impact on the image's semantic content, depending on the task being considered. For example, illumination or image exposure changes should not affect an object classification pipeline. To build systems invariant to such semantically-irrelevant changes, two high-level approaches are commonly used: *i)* normalization and *ii)* data augmentation.

Normalization eliminates a source of variation altogether by bringing input images to a single canonical representation. For illumination, this corresponds to equalization algorithms such as CLAHE [7]; to compensate camera movements, digital, optical, and mechanical stabilization methods exist [8]. Image stabilization has been widely deployed in computer vision, including in MAV applications: e.g. the Parrot Bebop 2 uses digital stabilization, whereas the DJI Phantom 2 relies on a mechanical gimbal. Its main drawback is cost at runtime, whether in computational power or in mechanical complexity, which, to this date, prevents adoption on nano-sized MAVs.

Alternatively, CNNs can learn arbitrary invariants when they are sufficiently represented in the training data. As real-world data is typically scarce, especially in robot applications, deep learning pipelines make extensive use of data augmentation for this purpose: available data is artificially in-

*Email address(es): elia.cereda@idsia.ch

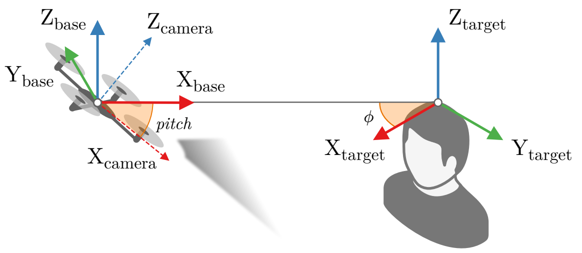

Figure 1: Human pose estimation task and reference frames.

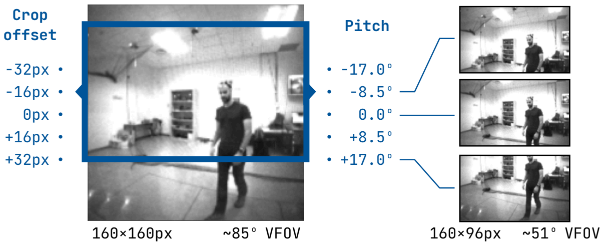

Figure 2: Pitch augmentation technique and example images.

creased by randomly perturbing copies of each training instance with transformations that do not affect the corresponding target variables. Being applied only at training time, augmentation is ideal for applications with tight resource constraints at inference time, such as nano-drones.

A rich data augmentation literature exists [9]. Generic photometric adjustments that simulate illumination changes are some of the most widely applied, together with small-scale geometric perturbations [10]. Task-specific augmentation strategies have also been proposed: *domain randomization* to improve generalization to unseen environments [11, 12]; *view synthesis* to increase the density of camera poses for visual localization and semantic segmentation [13, 14]; meta-learning approaches allow automatic task-specific fine-tuning of augmentations [15]. Our proposed *pitch augmentation* technique can be seen as a constrained version of view synthesis, which deals exclusively with camera orientation changes along the pitch axis.

Directly-measurable sources of variations allow an additional, orthogonal strategy: the current state can be fed as an explicit additional input to the CNN (vision-state fusion). In robotic control-oriented tasks and in egocentric spatial perception tasks, where the output refers to the system itself, this is an established practice. Abbeel et al. [16] use a drone's linear and angular velocities as the only inputs of a learned model that performs acrobatic maneuvers; more recent control approaches integrate the state and visual features in a single neural network [17]. Egocentric visual-inertial state estimation uses camera images and the robot's velocities and orientations measured by an inertial measurement unit (IMU), either through a sequence-to-sequence recurrent neural network that fuses the two input sequences and produces the sequence of egocentric poses [18, 19], or through a feed-forward architecture that takes an inertial initial motion estimation and refines it using the visual feed [20].

In this work, instead, we consider human relative pose estimation: an *allocentric* spatial perception task, whose output is external to the robot. Current CNN-based approaches for this class of problems rely exclusively on visual inputs [4, 21] but, for them too, recent advances show benefits when exploiting additional state information [22]. We evaluate a modified CNN architecture that receives the current pitch as additional input from the drone's onboard state estimation.

3 METHOD

3.1 Application

The reference application for our work is human pose estimation aboard the Crazyflie 2.1[1] 27-gram MAV by Bitcraze, coupled with the AI-deck pluggable extension board which provides a Himax HM01B0 gray-scale QVGA camera and a GreenWaves Technologies GAP8 ultra-low power multi-core system-on-chip (SoC). We leverage the *PULP-Frontnet* convolutional neural network, a field-proven model which takes one gray-scale 160×96px image and estimates the pose of the human subject relative to the drone (see Figure 1). The model produces four independent regression outputs, which represent the Cartesian coordinates of the subject's position in 3D space (x, y, z) w.r.t. the drone's horizontal $\{\text{base}\}$ frame and the subject's relative orientation w.r.t. the drone's yaw, ϕ. In this work, we focus on the position components (x, y, z).

To train our models, we employ ground-truth data collected in rooms equipped with a motion capture (mocap) system. Images streamed from the drone's camera are stored together with the corresponding drone's and subject's absolute poses, as recorded by the mocap system, and the drone's attitude, computed by onboard state estimation.

3.2 Pitch augmentation

During flight, the quadrotor maneuvers by continuously adjusting its pitch to generate forward/backward thrust. Without any kind of digital or mechanical stabilization (i.e., normalization), these attitude changes directly affect the geometry of the drone camera images. Pitch augmentation addresses this problem by synthetically enhancing the pitch range represented in the CNN's training set, at the expense of a reduced vertical field of view (VFOV). As shown in Figure 2, we acquire 160×160px images from the Himax camera with a 85° VFOV. By cropping 160×96px sub-regions at varying vertical offsets, we can simulate 65 different images with a 51° VFOV in a ±17° range of relative pitch compared to the original image. In other words, the top 96-row region of an image shows approximately the same scene at a $-17°$ pitch compared to the central 96-row region.

This simple approximation neglects the perspective and radial distortion components of the real geometric transfor-

[1] https://bitcraze.io/products/crazyflie-2-1/

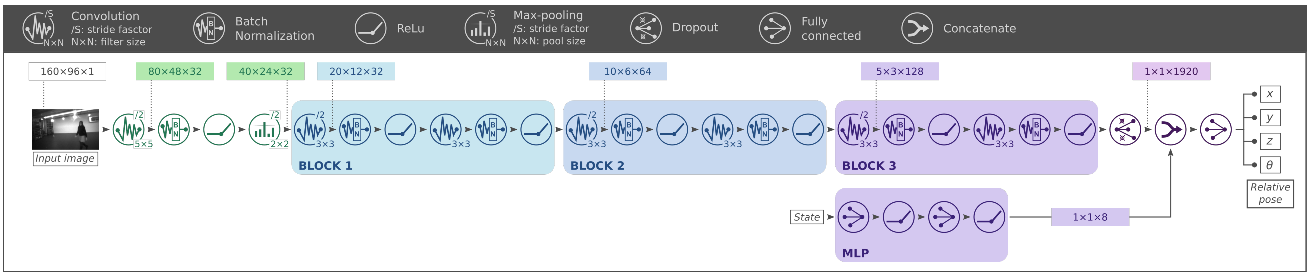

Figure 3: PULP-Frontnet CNN architecture [4], extended with the multi-layer perceptron branch for the state input.

mation imposed by a pitch change, but has the advantage of generating augmented sub-regions that are always entirely contained in the original image.

3.3 Vision-state fusion

We also propose an orthogonal approach to deal with changes in drone attitude by feeding to the model the current pitch in addition to the image; we expect that the CNN will exploit this information to explicitly account for pitch changes. This takes advantage of information already available on board, computed by the state estimation system from measurements of the inertial measurement unit (IMU).

We compare the four models that result from combining pitch augmentation applied to the training data (referred from here on as models *w/* or *w/o pitch aug*) and the additional pitch input fed to the CNN (i.e., *stateless* and *stateful* models). When evaluating models trained with both pitch augmentation and stateful input, care must be taken to ensure that the state input remains consistent after the augmentation: we sum the original image's ground-truth pitch with the synthetic pitch angle that corresponds to the random region selected by the augmentation. The resulting angle, which is then fed to the model, is the pitch that would produce a real image with the same orientation as the augmented one.

4 Experimental setup

Our model architecture is based on PULP-Frontnet [4], a 9-layer CNN tuned for the GAP8 SoC on the target nano-drone. We follow the original execution strategy when deploying the trained models: 8-bit quantized arithmetic for all CNN operations and automatically-generated tiling code, respectively to compensate for the lack of floating-point units on the GAP8 and to efficiently exploit its explicitly-managed memory hierarchy and 8-core parallel cluster.

For our vision-state fusion experiments, we extend our architecture to receive the drone's current pitch from the state estimation as an additional scalar input. As described in [22], we pre-process the state input with a small multi-layer perceptron (MLP), composed of two 8-unit fully connected layers (FC) interleaved by ReLU non-linearities, before concatenating it to the CNN's main FC layer. The resulting architecture (see Figure 3) has a negligible increase in inference-time memory and computation requirements, respectively +120 bytes and +140 multiply-accumulate operations (MACs), compared to $300\,\mathrm{kB}$ and $14\,\mathrm{MMACs}$ total for the original PULP-Frontnet architecture. We deploy the MLP branch with sequential software-emulated floating-point arithmetic, which accounts for just 0.5% of the total workload ($\sim$ 20k clock cycles).

Data for our experiments is collected in two different mocap-equipped indoor laboratories and includes a combination of static samples with the drone fixed horizontally on a wheeled cart (20%) and in-flight samples with the drone controlled by a human pilot (80%). In total, we record 12k camera frames, with associated ground-truth poses and estimated attitudes, from sessions with 17 human subjects of different age, height, ethnicity, and clothing. Three subjects (4.7k samples) form our test set, while the remaining 14 subjects (7.3k) are split into training (90%) and validation (10%) sets.

In our strategy, we augment each training sample 10 times, as an offline pre-processing step, performing a number of standard photometric data augmentations to promote robustness of our model to illumination changes – exposure, gamma correction, dynamic range adjustments, addition of Gaussian noise, and blurring – followed by vignetting, and horizontal flipping with 50% probability (shown in Figure 4). When enabled, our proposed pitch augmentation is also applied during this step, before all other augmentations. On the training and validation sets, we uniformly select a random synthetic pitch from the $\pm 17°$ range. For one experiment, we perform pitch augmentation also at test time – without the other augmentations – in this case generating every possible synthetic pitch for each test image (65 copies). Finally, we discard samples in which the subject is outside the field of view, obtaining the distributions of camera pitch values shown in Figure 5. We observe that pitch augmentation has a strong regularizing effect on the distribution, removing the heavy bias towards pitch $0°$ of the real data – i.e., w/o pitch aug in Figure 5.

Training is performed for 100 epochs using the Adam optimizer with learning rate 10^{-3} to minimize the L1 loss of the relative pose. At the end of training, the model checkpoint that reached the best performance on the validation set is selected for evaluation on the test set.

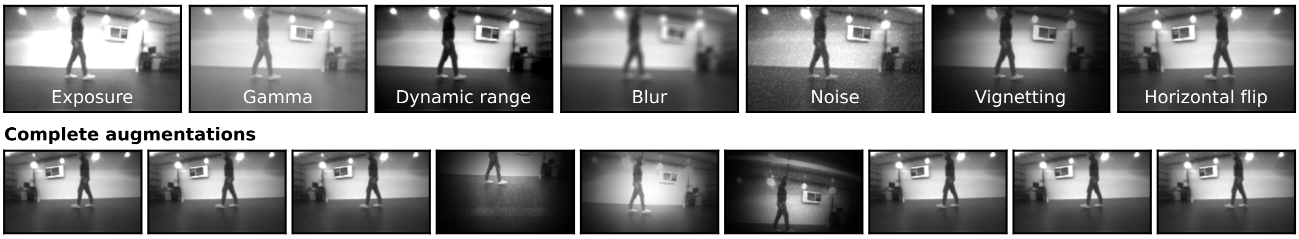

Figure 4: Individual photometric data augmentations (top). Ten images produced by the entire augmentation pipeline (bottom).

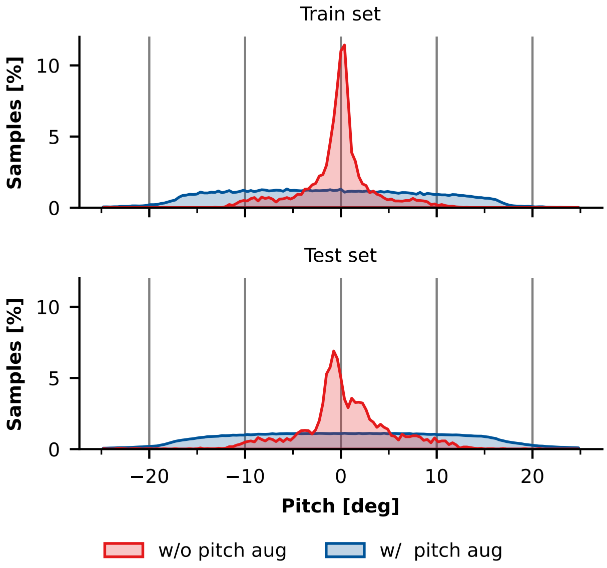

Figure 5: Distributions of camera pitch in the datasets.

5 Experimental results

In the following, we first evaluate the performance of our models in offline experiments on the test set. We train five instances of each model and measure their regression performance using the coefficient of determination R^2, a standard adimensional metric that represents the fraction of variance in the target variable explained by the model[2]. An $R^2 = 1.0$ corresponds to a perfect regressor, while a dummy regressor that always outputs the mean of the test data achieves $R^2 = 0.0$. Models can perform arbitrarily worse than the dummy regressor, leading to negative R^2 scores. The R^2 score is closely related to another standard regression metric, the mean squared error (MSE)[3]. Unlike the MSE, the R^2 score quantifies the quality of the regressor independently of the variance of the target variable, and is therefore suitable for comparing regression performance on different variables. To further validate our results, we also evaluate the entire closed-loop system, deploying the models aboard the nano-drone and evaluate their in-flight behavior.

5.1 Offline experiments

We show the four models' regression performance in terms of R^2 score on the test set in Figure 6-A. The *stateless* model *w/o pitch aug* (hollow red model) represents our baseline. Introducing pitch augmentation alone (hollow blue) corresponds to the original PULP-Frontnet model, trained and evaluated on our new data. We see a strong positive effect on the performance of all three output components (x, y, z) when compared to the baseline, with an increase in median R^2 of respectively $+0.13$, $+0.12$, and $+0.09$. The original PULP-Frontnet is trained exclusively on static training data at 0° pitch, with pitch augmentation as the only source of pitch variability. In contrast, we show in this work that pitch augmentation is strongly beneficial even when a vast majority of training data is collected in flight with real pitch variations.

On the other hand, introducing the state input alone (filled red) has a weak effect compared to the baseline. Only on z we see a median R^2 increase of $+0.07$, while x and y are much closer ($+0.02$ and $+0.00$). A similarly weak effect is visible when the state input is introduced on top of pitch augmentation (filled blue vs. hollow blue): on z we see a median R^2 increase of $+0.06$, while x and y are not affected. Overall, pitch augmentation strongly improves all three output components, while the state input noticeably benefits only z (intuitively, the component on which pitch has the strongest visual impact). The two techniques can be applied simultaneously, with the respective benefits compounding to reach the top performance.

We further analyze the four models' regression performance across a wide range of pitch angles by applying test-time pitch augmentation to the test set, as discussed in Section 4. Unsurprisingly, the overall regression performance in Figure 6-B decreases for all models compared to Figure 6-A. Those trained without pitch augmentation (red) in particular show a drop in median R^2 score of up to -0.33. Relative performance across the four model is unchanged compared to the raw test set. The nearly-uniform distribution imposed

[2] Defined as $R^2 = 1 - \sum_i (y_i - \hat{y}_i)^2 / \sum_i (y_i - \bar{y})^2$ with y_i the ground-truth output and $\hat{y}_i$ the model prediction for each test sample i, $\bar{y}$ the mean of ground-truth outputs.

[3] In fact, rewriting the formula above gives $R^2 = 1 - \mathrm{MSE}/\mathrm{MSE}_{dummy}$.

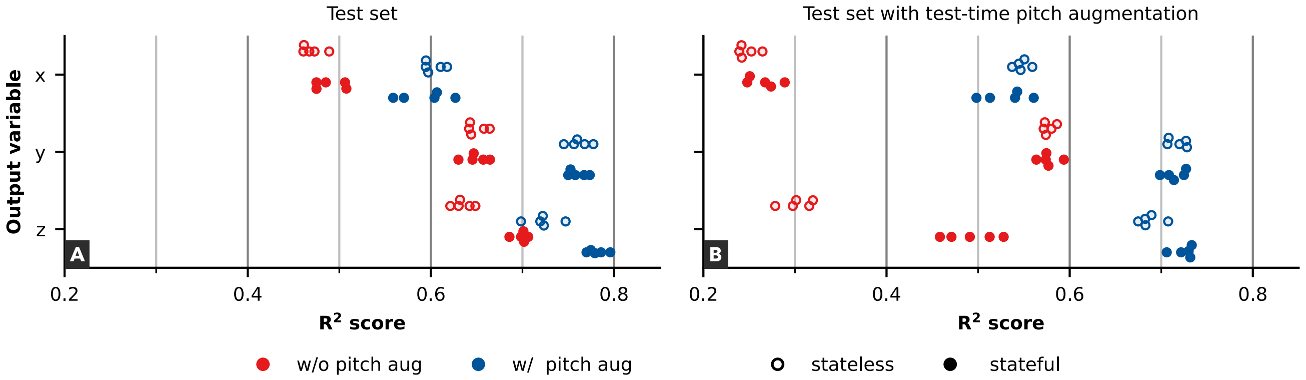

Figure 6: Overall regression performance in the offline experiments on the test set.

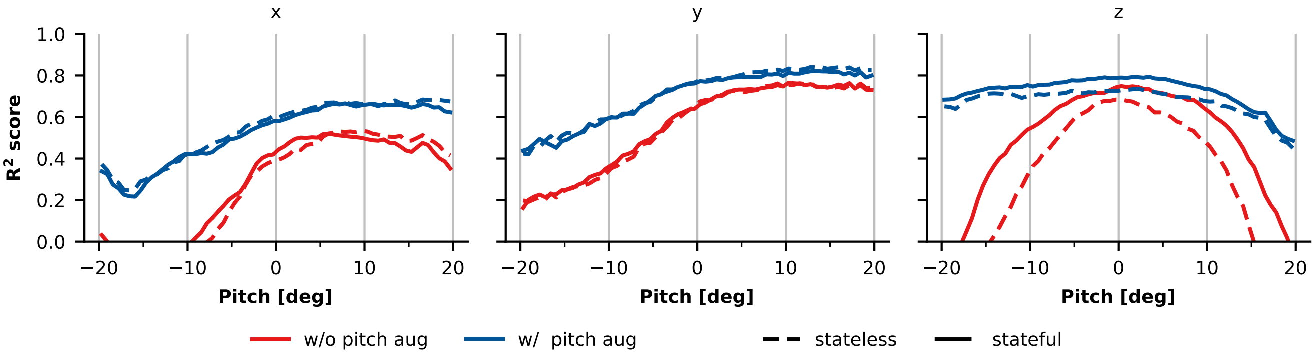

Figure 7: Regression performance on the test set with test-time pitch augmentation, broken down by drone pitch.

by pitch augmentation allows us to clearly visualize the regression performance broken down by pitch angle in Figure 7. Non-augmented models (red) exhibit a drastic drop at extreme pitch angles, especially on z, where performance falls to $R^2 = 0.0$ at ±15°. The performance of augmented models (blue) remains more stable across the entire range of pitch values: we attribute this to scarcity of real-world training data at extreme pitches, further confirming the effectiveness of pitch augmentation. On top of that, performance on x and y decreases at negative pitch angles (camera looking up), with pitch augmentation only partially compensating. Often, from visual inspection of the data, just the subject's head remains visible at strongly negative pitch, making accurate predictions harder.

5.2 In-field evaluation

In the second experiment, we focus on the two best-performing models, those trained with pitch augmentation (blue), and evaluate their behavior in the field when deployed aboard a closed-loop fully-autonomous Crazyflie nano-drone. We evaluate a combination of two scenarios and three distinct subjects, repeating each for five flights of ∼10 s per model, a total of 60 test flights. In both scenarios, the drone starts hovering at $0.5\,\mathrm{m}$ altitude and at a $\sim 6\,\mathrm{m}$ horizontal distance in front of the subject, then the onboard CNN inference is used to autonomously control the drone towards a target position $1.3\,\mathrm{m}$ in front of the person at eye level. Forward velocity is limited to a maximum of $1.2\,\mathrm{m/s}$.

In the first scenario, the subjects stand still for the entire duration of the test, while the drone climbs from the hovering altitude up to their height (1.65–$1.85\,\mathrm{m}$) and simultaneously moves forward towards them. We are thus stressing the models' x and z predictions. Figure 8-A shows the drone's distance from the target in the horizontal plane over time under this scenario: both models achieve almost identically good behavior, converging to the target with almost no error nor oscillations. Figure 8-B, on the other hand, shows the difference (delta) between drone and target altitudes over time. Here, the models' behaviors differ noticeably: while the stateful model firmly converges to the desired position, the stateless model cannot reach it and overshoots instead.

The second scenario is identical to the first at the beginning. Then, after 6 s from the start, the subjects are instructed to kneel down. This strongly challenges dynamic z predictions, as the drone has almost reached the target position $1.3\,\mathrm{m}$ in front of the subjects when they start kneeling.

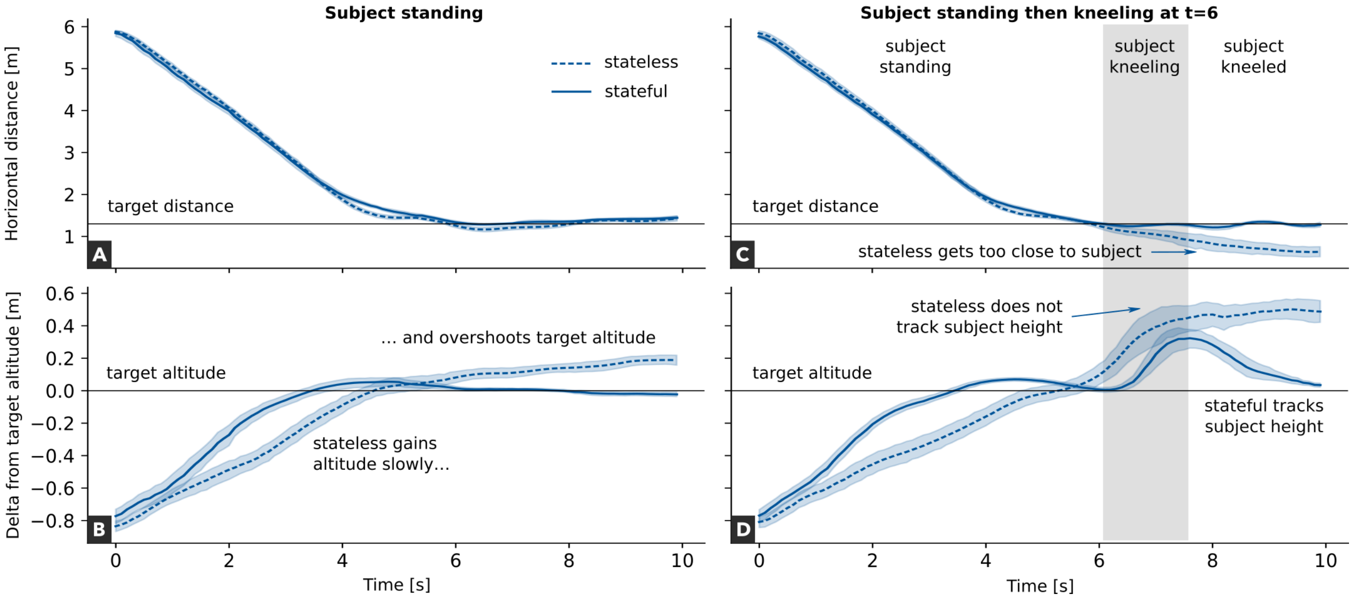

Figure 8: In-field behavior of the stateless and stateful models trained with pitch augmentation.

This means the person's downward motion corresponds to a considerable movement in image space, giving the drone little time to react before losing the target outside the field of view. Under this scenario, we see the stateless model struggling to follow subjects as they kneel, both in the horizontal plane (Figure 8-C) and in altitude (Figure 8-D). In contrast, the stateful model both precisely converges in front of the standing subjects ($t = 6$s), and accurately follows them as they kneel ($t = 6$s to 7.5s), keeping the correct horizontal distance and descending to reach the new target altitude.

From this experiment, we conclude that the state input (pitch) leads to significantly improved performance in the field. While little difference between the stateless and stateful models was seen in the offline experiments, we show that the stateful model's in-flight closed-loop behavior significantly outperforms its stateless counterpart.

6 Conclusion

We considered a perception problem for MAVs solved using a regression CNN; to enforce robustness to image variations due to drone pitch, we proposed and quantitatively evaluated an approach based on data augmentation, achieving improvements in median R^2 score of up to +0.15; we further verified that providing explicit information about the drone pitch to the model yields improvements both in offline and in-field control experiments.

Acknowledgements

This work was partially supported by the Secure Systems Research Center (SSRC) of the UAE Technology Innovation Institute (TII).

References

[1] Guanya Shi, Wolfgang Hönig, Yisong Yue, and Soon-Jo Chung. Neural-Swarm: Decentralized close-proximity multirotor control using learned interactions. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3241–3247, 2020.

[2] Wenda Zhao, Jacopo Panerati, and Angela P. Schoellig. Learning-based bias correction for time difference of arrival ultra-wideband localization of resource-constrained mobile robots. *IEEE Robotics and Automation Letters*, 6(2):3639–3646, 2021.

[3] Daniele Palossi, Francesco Conti, and Luca Benini. An open source and open hardware deep learning-powered visual navigation engine for autonomous nano-UAVs. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 604–611. IEEE, 2019.

[4] Daniele Palossi, Nicky Zimmerman, Alessio Burrello, Francesco Conti, Hanna Müller, Luca Maria Gambardella, Luca Benini, Alessandro Giusti, and Jérôme Guzzi. Fully onboard AI-powered human-drone pose estimation on ultra-low power autonomous flying nano-UAVs. *IEEE Internet of Things Journal*, 9(3):1913–1929, 2021.

[5] Naotoshi Abekawa, Elisa R Ferrè, Maria Gallagher, Hiroaki Gomi, and Patrick Haggard. Disentangling the visual, motor and representational effects of vestibular input. *Cortex*, 104:46–57, 2018.

[6] Elisa R Ferrè, Adrian JT Alsmith, Patrick Haggard, and Matthew R Longo. The vestibular system modulates the contributions of head and torso to egocentric spatial judgements. *Experimental Brain Research*, 239(7):2295–2302, 2021.

[7] Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B. Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39(3):355–368, 1987.

[8] Marcos Roberto e Souza, Helena de Almeida Maia, and Helio Pedrini. Survey on digital video stabilization: Concepts, methods, and challenges. *ACM Comput. Surv.*, 55(3), feb 2022.

[9] Connor Shorten and Taghi Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, 07 2019.

[10] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc., 2020.

[11] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[12] Elia Cereda, Marco Ferri, Dario Mantegazza, Nicky Zimmerman, Luca M. Gambardella, Jérôme Guzzi, Alessandro Giusti, and Daniele Palossi. Improving the generalization capability of DNNs for ultra-low power autonomous nano-UAVs. In *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 327–334, 2021.

[13] Yiming Wan, Wei Gao, Sheng Han, and Yihong Wu. Boosting image-based localization via randomly geometric data augmentation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 688–692, 2020.

[14] Joris Guerry, Alexandre Boulch, Bertrand Le Saux, Julien Moras, Aurelien Plyer, and David Filliat. SnapNet-R: Consistent 3D multi-view semantic labeling for robotics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[15] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *European Conference on Computer Vision*, pages 566–583. Springer, 2020.

[16] Pieter Abbeel, Adam Coates, and Andrew Y Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010.

[17] Elia Kaufmann, Antonio Loquercio, Rene Ranftl, Matthias Mueller, Vladlen Koltun, and Davide Scaramuzza. Deep drone acrobatics. In *ROBOTICS: SCIENCE AND SYSTEMS XVI*, 2020.

[18] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. VINet: Visual-inertial odometry as a sequence-to-sequence learning problem. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017.

[19] Liming Han, Yimin Lin, Guoguang Du, and Shiguo Lian. DeepVIO: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6906–6913, 2019.

[20] Sudeep Pillai and John J. Leonard. Towards visual egomotion learning in robots. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5533–5540, 2017.

[21] Elia Kaufmann, Mathias Gehrig, Philipp Foehn, René Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Beauty and the beast: Optimal methods meet learning for drone racing. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 690–696. IEEE, 2019.

[22] Elia Cereda, Stefano Bonato, Mirko Nava, Alessandro Giusti, and Daniele Palossi. Vision-state fusion: Improving deep neural networks for autonomous robotics, 2022. [online pre-print].