3D CNN and Multi-Head Attention to Relative Camera Position

Nilda G. Xolo-Tlapanco and Jose Martinez-Carranza *
Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, México

http://www.imavs.org/

ABSTRACT

Calculating the relative position between images in Unmanned Aerial Vehicles (UAVs) is a core component in tasks such as visual odometry, SLAM, and state estimation. It enables the UAV to estimate its movement between frames using onboard sensors. In this paper, we present a spatio-temporal regression architecture combining 3D CNNs and masked attention mechanisms. The model takes a sequence of image frames and corresponding Inertial Measurement Unit (IMU) data for each frame as input and outputs the estimated relative position in meters between the images. The inclusion of IMU measurements is critical for improving robustness to motion blur, low-texture environments, and rapid maneuvers, as it provides complementary information about the UAV's linear acceleration and angular velocity. The CNN layers extract compact spatio-temporal features from the video input, while the multi-head attention layer captures temporal dependencies and contextual relations across time. The final Multi-Layer Perceptron (MLP) regresses these fused representations into a relative position estimate. We demonstrate the effectiveness of our method on the TII Drone Racing and UZH-FPV datasets.

1 INTRODUCTION

Accurate estimation of the relative camera pose between consecutive frames is a fundamental problem in computer vision and robotics, with applications in autonomous UAV navigation, visual odometry, augmented reality, and simultaneous localization and mapping (SLAM). Estimating how a camera moves through an environment using only visual data enables devices to operate in GPS-denied or dynamic environments, a crucial capability for lightweight, agile aerial robots. However, relying solely on visual input can be challenging in scenes with motion blur, low texture, or sudden accelerations [1]. To address this, inertial measurements from an Inertial Measurement Unit (IMU) provide complementary motion cues, capturing high-frequency information about the UAV's linear acceleration and angular velocity. When fused with visual data, IMU information improves pose estimation accuracy and temporal consistency, particularly in fast or ambiguous motion scenarios [2].

Traditionally, relative pose estimation has been based on geometry-based methods such as feature matching, essential matrix decomposition, and epipolar geometry [3, 4, 5]. Although effective under ideal conditions, these approaches are sensitive to image noise, motion blur, dynamic objects, and low-texture scenes (all common in UAV flight scenarios). More recently, learning-based approaches have shown promise in extracting robust representations from raw images, enabling the end-to-end estimation of camera motion without the need for explicit feature extraction or geometric modeling [6], as well as the use of only IMU information to calculate the relative pose and orientation [7, 8]. However, most existing learning methods rely on 2D convolutional networks, which fail to fully capture the temporal dynamics and motion continuity inherent in sequential image data.

To address these limitations, we propose a novel deep learning architecture that combines 3D Convolutional Neural Networks (3D CNNs) with a masked multi-head attention mechanism for robust and efficient estimation of relative camera motion with the help of inertial measurements from an IMU. The 3D CNN serves as a spatio-temporal feature extractor, capturing both appearance and motion patterns across frames, while the attention module models long-range dependencies and spatial-temporal interactions, improving the consistency and robustness of the motion estimation.

2 RELATED WORK

Traditional approaches for estimating the relative pose between two images rely heavily on geometric principles, such as epipolar geometry and Structure-from-Motion (SfM) pipelines. These methods typically involve detecting and matching feature points (like SIFT or ORB methods), estimating the fundamental or essential matrix, and decomposing it to recover relative rotation and translation (up to scale) [9]. While effective under controlled conditions, these techniques degrade significantly in low-texture regions, under motion blur, or when the scene contains dynamic elements, scenarios common in UAV operations.

With the rise of deep learning, several methods have been proposed to estimate camera motion directly from raw image data. PoseNet [10] introduced a convolutional architecture for regressing the absolute camera pose from a single image, later extended to use temporal information. DeepVO [6],

*Email address: carranza@inaoep.mx

http://www.imavs.org/

SfM-Net [11], and SelfVIO[12] focused on estimating relative motion using Recurrent Neural Networks (RNN) or convolutional architectures on image pairs or sequences. These methods bypass traditional feature extraction but often rely on 2D CNNs, which may not fully capture temporal dynamics across frames.

3D convolutional neural networks have been widely adopted in video classification [13] and action recognition [14] due to their ability to model spatio-temporal dependencies. Models like C3D [15] and I3D [16] extract features from both spatial and temporal dimensions, making them suitable for motion understanding. In the context of pose estimation, 3D CNNs offer the advantage of learning motion cues directly from short video clips. However, their use in relative pose estimation, particularly for UAVs, remains underexplored.

The introduction of Transformers [17] and their adaptation to vision tasks (ViT [18], TimeSformer [19]) have shown that self-attention can be highly effective in modeling long-range dependencies and contextual relationships in image and video data. In pose estimation, attention-based models such as ConvLSTM [20] and DROID-SLAM [21] incorporate attention layers to integrate spatial and temporal information between image sequences. These methods demonstrate the potential of attention mechanisms to enhance robustness, especially under challenging conditions.

For aerial robotics, estimating relative pose is especially critical due to limited access to GPS, high-speed maneuvers, and environmental variability. Methods like VINS-Mono [22] and ORB-SLAM3 [23] provide tightly coupled visual-inertial and SLAM-based solutions. Learning-based approaches, such of Xu et al. [24], demonstrate that compact convolutional architectures can effectively handle large disparities and motion blur in a downward-facing camera, particularly when aided by IMU signals. However, these models typically operate in a frame-to-frame setting and remain limited in capturing longer-term temporal dependencies.

Our work addresses these challenges by combining 3D CNNs for spatio-temporal motion encoding with multi-head attention mechanisms that incorporate both visual and inertial data. Specifically, we integrate IMU measurements (linear accelerations and angular velocities) as additional input streams into the attention layers, enabling the model to learn cross-modal dependencies and temporal patterns that extend beyond what vision alone can capture. This fusion improves the model's ability to disambiguate motion in scenes with textureless surfaces, motion blur, or fast UAV maneuvers. By attending jointly to visual features and inertial cues, our architecture produces more accurate and robust relative pose estimates tailored to the dynamics of agile aerial platforms.

3 METHODOLOGY

This section presents the proposed architecture for estimating the relative camera translation between consecutive frames. The network combines 3D Convolutional Neural Networks (3D CNNs) and a masked multi-head attention mechanism to model both local spatio-temporal patterns and long-range dependencies across image sequences. The output is a 3-dimensional vector corresponding to the relative displacement (x,y, and z) of the camera. Given a sequence of consecutive RGB images $I_t, I_{t+1}, \ldots, I_{t+n}$ and IMU data $IMU_t, IMU_{t+1}, \ldots, IMU_{t+n}$ representing the angular velocity $(\omega_x, \omega_y, \omega_z)$ and linear acceleration (a_x, a_y, a_z) of each image, the objective is to estimate the relative translation $t_{t\to t+1} \in R^3$ between frames I_t and I_{t+1}, assuming a known or negligible rotation. This formulation is useful in UAV applications such as visual odometry, drone racing, and inspection, where accurate estimation of egomotion is crucial. The proposed architecture consists of four main stages (Figure 1):

- **Spatio-temporal feature extraction via 3D CNN:** A stack of 3D convolutional layers processes the input sequence $X \in \mathbb{R}^{B\times C\times T\times H\times W}$, where B is the batch size, C the number of channels, T the number of frames, and $H \times W$ the image resolution (set at 122×122 in our experiments). The 3D convolutions jointly encode appearance and temporal motion patterns. After three convolutional blocks (with batch normalization and ReLU activations), the output is reshaped into a sequence of feature vectors of size 128, yielding $X_{vis} \in \mathbb{R}^{B\times L\times 128}$, where $L = C' \times H' \times W'$ is the flattened spatio-temporal dimension of the last 3D convolution, in this case different from the original dimension.

- **IMU feature encoding:** Raw IMU signals $(\omega_x, \omega_y, \omega_z, a_x, a_y, a_z)$ over a clip of T frames are stacked into a vector of dimension $6T$ (i.e., 36 when $T = 6$). A two-layer Multi-Layer Perceptron (MLP) projects this input to a 128-dimensional embedding, $X_{imu} \in \mathbb{R}^{B\times 128}$. This representation is then repeated along the sequence length L, resulting in $X_{imu} \in \mathbb{R}^{B\times L\times 128}$, to match the visual features.

- **Temporal attention via masked multi-head attention:** The visual features X_{vis} and IMU features X_{imu} are concatenated along the feature dimension, producing $X_{fused} \in \mathbb{R}^{B\times L\times 256}$. This sequence is passed to a masked multi-head self-attention module that:
  - Applies multiple attention heads to compute relationships between all pairs of feature tokens,
  - Uses an optional mask to restrict attention to valid temporal regions,
  - Enables the model to reason over global temporal dependencies and non-local motion patterns.

- **Regression head via MLP:** The attention-enhanced sequence is pooled via global average pooling over L, resulting in a fixed-size representation $\in \mathbb{R}^{B\times 256}$. A

http://www.imavs.org/

three-layer MLP regresses the final translation vector (x, y, z).

The network is trained using supervised learning with ground truth relative translations, which are calculated with the ground truth position and orientation of the dataset.

Given two poses at timestamps t_i and t_{i+1}, each pose consists of a position vector $\mathbf{t} \in \mathbb{R}^3$ and a unit quaternion $\mathbf{q} \in \mathbb{R}^4$ representing orientation.

1. Homogeneous transformation matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}(\mathbf{q}) & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$$

2. Relative transformation: Let $\mathbf{T}_1$ and $\mathbf{T}_2$ be the transformation matrices at times t_i and t_{i+1}, respectively. The relative transformation is:

$$\mathbf{T}_{\text{rel}} = \mathbf{T}_1^{-1} \cdot \mathbf{T}_2$$

3. Output format:

$$\mathbf{T}_{\mathbf{rel}} = \begin{bmatrix} \mathbf{R}(\mathbf{q_{rel}}) & \mathbf{t_{rel}} \\ \mathbf{0} & 1 \end{bmatrix}$$

From this matrix, we extract the position and orientation to use separately.

Reconstructing global trajectory from relative poses Given a sequence of relative poses $\mathbf{t}^i_{\text{rel}}$ predicted by our network and relative orientations by the ground truth $\mathbf{q}^i_{\text{rel}}$ for $i = 0, 1, \ldots, N-1$, we reconstruct the global trajectory using the following procedure.

1. Relative transformation matrix: Each relative pose is converted to a homogeneous transformation:

$$\mathbf{T}^i_{\text{rel}} = \begin{bmatrix} \mathbf{R}(\mathbf{q}^i_{\text{rel}}) & \mathbf{t}^i_{\text{rel}} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3)$$

2. Initialization: Start at the origin with identity transformation:

$$\mathbf{T}_0 = \mathbf{I}_{4\times 4}$$

3. Recursively compute global poses

$$\mathbf{T}_{i+1} = \mathbf{T}_i \cdot \mathbf{T}^i_{\text{rel}}$$

For the training, we calculate the loss with Mean Squared Error (MSE) in meters between prediction and ground truth translation t_{gt}:

$$L_{\text{pose}} = \|t_{\text{pred}} - t_{\text{gt}}\|^2 \quad (1)$$

4 EXPERIMENTS AND RESULTS

We evaluate our model on the TII Drone Racing [25] and UZH-FPV [26] datasets. The TII Drone Racing contains RGB images with illumination changes from fast (>21m/s) and aggressive quadrotor flight and the UZH-FPV dataset includes RGB images from indoor and outdoor scenes of agile flight.

The experiments were carried out on a computer equipped with an Intel Core i5-9400F, 16GB of RAM, and an NVIDIA GeForce RTX 4070 Super GPU. Due to the nature of our network, it is necessary to determine the optimal value for T (the number of frames required to subtract enough information to estimate the relative position). Therefore, we evaluate the network with T values of 2, 4, 6,8, and 10 on the UZH-FPV dataset using sequences Indoor forward facing 03, 05, and 06 for training, 07 and 09 for validation, and 10 for testing. These sequences were selected because they represent different flight trajectories and visual conditions within the same forward-facing setup. For example, training sequences (03, 05, 06) contain a mix of straight and turning maneuvers, providing sufficient variability for the model to learn motion patterns. Validation sequences (07, 09) include faster translational movements and different illumination, which are useful to monitor overfitting. Finally, sequence 10 was reserved for testing as it features a distinct trajectory not seen during training, ensuring an unbiased evaluation of generalization.

We analyze the training loss over 100 epochs. The T=10 setting is the most unstable; however, in general, the loss decreases consistently throughout training. To determine the best T value, we compare the MSE of each model for the test sequence in Table 1, with the best result obtained for T=6.

With the optimal value of T fixed at 6, we train the final model on the following UZH-FPV sequences to evaluate its effectiveness:

- Training set: Indoor forward-facing: 06, 07, 09, Outdoor forward-facing: 03, Indoor 45° downward-facing: 02, 04, 09, and Outdoor 45° downward-facing: 01
- Validation set: Outdoor forward-facing: 01 and Indoor 45° downward-facing 12.
- Testing set: Indoor forward-facing: 10, Outdoor forward-facing: 05 and Indoor 45° downward-facing: 14

With a total of 15,010 images for training, 3,253 for validation, and 3,065 for testing. In the case of the TII Drone racing dataset we use the following sequences:

- Training set: 01, 05, 08, 09, 11, and 12.
- Validation set: 02, 03, 06, and 07.
- Testing set: 04 and 10.

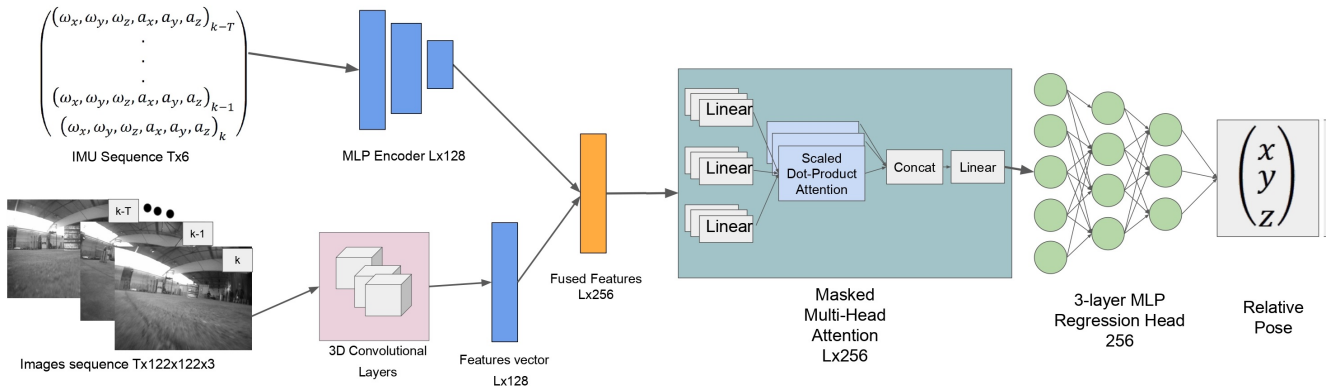

Figure 1: Overview of the proposed architecture. The model consists of four stages: (1) spatio-temporal feature extraction via a 3D CNN that encodes motion cues across frames, (2) IMU feature encoding with a lightweight MLP, (3) temporal attention using a masked multi-head attention mechanism to capture long-range dependencies, and (4) a regression head via MLP to estimate the translation vector (x, y, z), where T is the number of frames and L the new dimension of the last 3D convolution.

http://www.imavs.org/

With a total of 34,882 images for training, 23,087 for validation, and 12,008 for testing. We prepare two networks for comparison, the first and main one, 3D CNN-IMU Attention described in the methodology, and 3D CNN-Attention, where the IMU information is excluded. In the latter, the input for the multi-head attention consists only of the features provided by the 3D convolutional layers, allowing us to evaluate the importance of the IMU data.

T value	MSE
2	1.123
4	1.216
6	**0.919**
8	1.642
10	1.403

Table 1: MSE for each value of T (2,4,6,8, and 10) in testing trajectories.

We reconstruct the global position predicted by each network and present it in Figures 2 and 3, corresponding to the training and testing trajectories, respectively. Once the predictions are obtained, we compute the MSE for each trajectory and compare them with the ground truth (Table 2), where the best values are highlighted in black.

As can be observed, the network without IMU information performs better on the training trajectories—i.e., those it has already seen—compared to unseen trajectories used in validation and testing, where IMU information appears to improve performance in unfamiliar environments.

4.1 Ablation Study

To assess the importance of each component of the network, we conduct experiments with different variations of our approach:

- 3D CNN-Attention with IMU information (3D CNN-Att IMU).
- 3D CNN-Attention without IMU information (3D CNN-Att).
- 3D CNN without Attention (3D CNN without Att).
- 3D CNN with IMU information and without Attention (3D CNN IMU without Att).
- CNN with Attention without IMU (CNN- Att).

In the case of the 2D CNN baseline, to emulate the temporal encoding of the 3D CNN, we concatenate the 6 consecutive input frames into a 3×2 grid, forming a single 224× 224 image. This strategy allows the 2D CNN to process multiple frames at once, although the temporal relationships must be inferred indirectly from the spatial arrangement.

These experiments were carried out only on the UZH-FPV dataset. Training and inference times for each variation are reported in Table 3 and the MSE for each trajectory are presented in Table 4.

As observed, CNN-Attention achieves the best performance in most training cases, but not in unseen environments, where other variations of the network perform better. The main disadvantage of the CNN-based architecture compared to 3D-CNN is the significantly longer training and inference time; the 3D CNN is 96% faster in training and 54% faster in inference, which is an important factor in real-time scenarios. Training a 3D CNN can be faster because temporal information is encoded natively through 3D convolutions, whereas a 2D CNN with a frame grid wastes capacity and training time trying to infer temporal relations from a spatial layout that doesn't naturally represent time, and in the case for inference. Furthermore, the attention module plays a critical role in the

http://www.imavs.org/

network, as removing it in "3D CNN without Att" variation leads to a noticeable drop in accuracy for most sequences, having the worst MSE scores.

Sequence	3D CNN-Att	3D CNN- IMU Att
	Train	
Indoor 06	**0.0609**	0.0836
Indoor 07	**0.0715**	0.1537
Indoor 09	**0.0162**	0.0374
Outdoor 03	**0.0883**	0.1444
Indoor 45° 2	**0.0297**	0.0420
Indoor 45° 4	**0.0227**	0.0333
Indoor 45° 9	**0.0209**	0.0388
Outdoor 45° 1	**0.0450**	0.0538
DR 01	**0.0083**	0.0114
DR 05	**0.0061**	0.0105
DR 08	**0.0057**	0.0075
DR 09	**0.0088**	0.0126
DR 11	**0.0093**	0.0128
DR 12	**0.0082**	0.0106
	Validation	
Outdoor 1	**1.3317**	1.7819
Indoor 45° 12	2.5218	**0.8214**
DR 02	0.0922	**0.0378**
DR 03	0.1029	**0.0241**
DR 06	0.2087	**0.0465**
DR 07	0.0983	**0.0218**
	Test	
Indoor 10	0.1775	**0.1711**
Outdoor 5	1.3651	**1.3574**
Indoor 45° 14	5.3050	**1.4855**
DR 04	0.2087	**0.0654**
DR 10	0.0983	**0.0347**

Table 2: MSE in meters of each sequence for our network with (3D CNN- IMU Att) and without (3D CNN-Attention) IMU information, the best values are highlighted in black.

Network	Training Time (s)	Inference Time (s)
3D CNN-Att IMU	61.60	1.481e-03
3D CNN-Att	61.06	1.332e-03
3D CNN without Att	61.24	**0.932e-03**
3D CNN IMU without Att	**59.72**	1.128e-03
CNN- Att	1532.51	2.895e-03

Table 3: Mean training time for 100 epochs and inference time for the testing sequences in seconds. The best values are highlighted in bold.

5 CONCLUSION

We presented a novel deep learning architecture for estimating relative camera pose, a critical task in computer vision and robotics, particularly in Unmanned Aerial Vehicles (UAVs), where onboard computational resources are often limited. Our method combines 3D Convolutional Neural Networks (3D CNNs) with a masked multi-head attention mechanism that fuses visual and inertial data, specifically inputs from a monocular camera and an Inertial Measurement Unit (IMU), both commonly available on UAV platforms. Through extensive experiments, we evaluated the effectiveness of our architecture under different configurations and demonstrated that incorporating inertial measurements improves generalization to unseen environments, reducing the MSE error in all test trajectories. Additionally, the use of 3D CNNs contributes to 96% faster in training and 54% faster in inference and more efficient inference by leveraging spatio-temporal motion cues.

For future work, we plan to evaluate the robustness of our method in real-world flight scenarios perform broader comparisons with state-of-the-art visual–inertial odometry and SLAM approaches, and extend the model to estimate full six degrees of freedom poses (translation and rotation) using additional regression heads.

ACKNOWLEDGEMENTS

The author Nilda G. Xolo-Tlapanco, thanks SECIHTI in Mexico for their scholarship with CVU number 1230561.

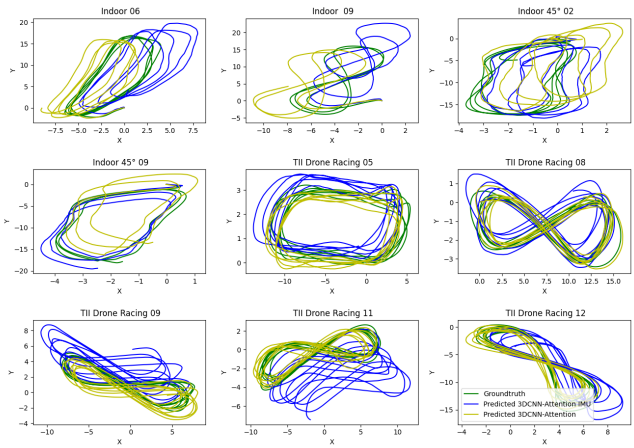

Figure 2: Reconstruction of some trajectories used for training, ground truth (green), predicted by 3D CNN-Attention IMU (blue), and predicted by 3D CNN-Attention (yellow).

REFERENCES

[1] Licong Zhuang, Xiaorong Zhong, Linjie Xu, Chunbao Tian, and Wenshuai Yu. Visual slam for unmanned aerial vehicles: Localization and perception. *Sensors*, 24(10), 2024.

Sequence	3D CNN-Att	3D CNN-Att IMU	3D CNN without Att	3D CNN IMU without Att	CNN- Att
			Train		
Indoor 06	0.0609	0.0836	1.3746	0.2071	**0.0283**
Indoor 07	0.0715	0.1537	1.5965	0.3428	**0.0597**
Indoor 09	0.0162	0.0374	0.3553	0.1351	**0.0134**
Outdoor 03	0.0883	0.1444	1.4994	0.3608	**0.0879**
Indoor 45° 2	**0.0297**	0.04204	0.3966	0.0764	0.0314
Indoor 45° 4	0.0227	0.0333	0.2639	0.0604	**0.0210**
Indoor 45° 9	**0.0209**	0.0388	0.4918	0.0528	0.0433
Outdoor 45° 1	0.0450	0.0538	0.3537	0.0662	**0.0305**
			Validation		
Outdoor 1	**1.3317**	1.7819	1.8531	0.0662	1.6686
Indoor 45° 12	2.5218	0.8214	0.8060	**0.0662**	1.2969
			Test		
Indoor 10	0.1775	**0.1711**	0.5431	0.3294	0.2086
Outdoor 5	1.36514	1.3574	1.3030	1.3754	**1.3313**
Indoor 45° 14	5.3050	1.4855	2.5718	**1.2046**	2.7904

Table 4: MSE in meters of each sequence for different variations of our network, the best values are highlighted in black.

http://www.imavs.org/

[2] Davide Scaramuzza and Zichao Zhang. *Visual-Inertial Odometry of Aerial Robots*, page 1–9. Springer Berlin Heidelberg, 2020.

[3] Hugh Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[4] Tserennadmid Tumurbaatar and Taejung Kim. Comparative study of relative-pose estimations from a monocular image sequence in computer vision and photogrammetry. *Sensors*, 19(8), 2019.

[5] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.

[6] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2043–2050, 2017.

[7] Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. Ionet: learning to cure the curse of drift in inertial odometry. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Press, 2018.

[8] Changhao Chen, Chris Xiaoxuan Lu, Johan Wahlström, Andrew Markham, and Niki Trigoni. Deep neural network based inertial odometry using low-cost inertial measurement units. *IEEE Transactions on Mobile Computing*, 20(4):1351–1364, 2021.

[9] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, March 2004.

[10] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015.

[11] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video, 2017.

[12] Yasin Almalioglu, Mehmet Turan, Muhamad Risqi U. Saputra, Pedro P.B. de Gusmão, Andrew Markham, and Niki Trigoni. Selfvio: Self-supervised deep monocular visual–inertial odometry and depth estimation. *Neural Networks*, 150:119–136, 2022.

[13] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[14] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.

[15] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal

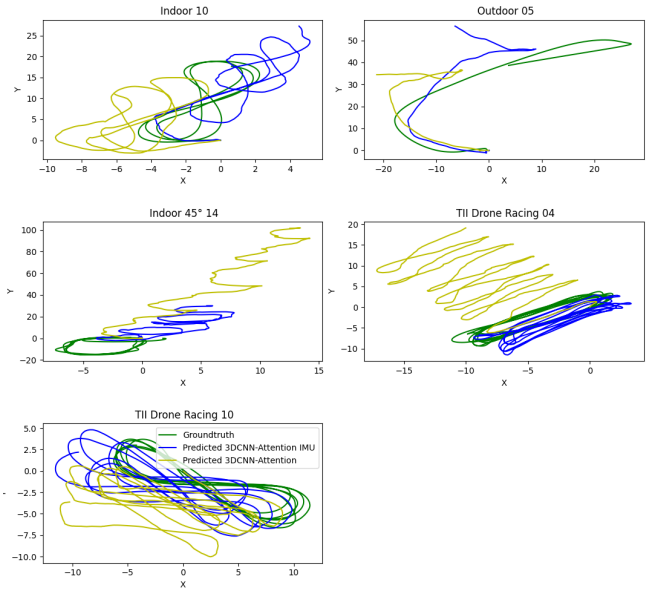

Figure 3: Reconstruction of the trajectories used for testing, ground truth (green), predicted by 3D CNN-Attention IMU (blue), and predicted by 3D CNN-Attention (yellow).

features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.

[16] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, July 2017.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[19] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

[20] Sangni Xu, Hao Xiong, Qiuxia Wu, and Zhiyong Wang. Attention-based long-term modeling for deep visual odometry. In *2021 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, 2021.

[21] Zachary Teed and Jia Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems*, 2021.

[22] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[23] Carlos Campos, Richard Elvira, Juan J. Gomez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[24] Yingfu Xu and Guido C. H. E. de Croon. Cnn-based ego-motion estimation for fast mav maneuvers. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7606–7612, 2021.

[25] Michael Bosello, Davide Aguiari, Yvo Keuter, Enrico Pallotta, Sara Kiade, Gyordan Caminati, Flavio Pinzarrone, Junaid Halepota, Jacopo Panerati, and Giovanni Pau. Race against the machine: A fully-annotated, open-design dataset of autonomous and piloted high-speed flight. *IEEE Robotics and Automation Letters*, 9(4):3799–3806, 2024.

[26] Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019.

http://www.imavs.org/