NeuroVIO: SNN Framework for Visual-Inertial Pose Estimation

Vidya Sudevan and Jorge Dias

***Abstract*— We present NeuroVIO, a hybrid end-to-end architecture that integrates conventional and spiking neural networks for multimodal visual-inertial odometry in underwater mobile robots. NeuroVIO addresses the need for using energy-efficient and accurate pose estimation methods in underwater mobile robots. In our approach, a CNN backbone extracts visual features from successive frames and converts them into time-encoded sequences, which are processed by adaptive leaky-integrate-and-fire neurons with learnable thresholds. Concurrently, inertial measurements are encoded via an SNN feature extractor. Fused features pass through a spike LSTM to capture temporal dependencies, and a spiking regression head predicts the six-dimensional pose vector. Evaluated on the AQUALOC dataset, the proposed NeuroVIO framework reduces the energy consumption by 80.4% relative to its non-spiking counterpart while preserving the pose estimation accuracy. The experimental results demonstrate that integrating neuromorphic paradigms into resource-limited marine robotics platforms enhances the autonomy of underwater robots in exploration tasks.**

I. INTRODUCTION

Accurate pose estimation is vital for navigation of autonomous underwater vehicles (AUVs) and remotely operated vehicles (ROVs). Underwater VIO is challenged by degraded visibility, low texture, dynamic lighting, and limited onboard computational resources [1]. Geometry-based methods such as MSCKF [2], OKVIS [2], and VINS [3] perform well in controlled settings but often fail underwater. Learning-based SOTA approaches exploit deep multimodal features, yet their high computational demand hinders real-time use on resource-limited robotic platforms [4].

Visual-inertial SLAM (viSLAM) fuses vision and IMU data, improving robustness in low-texture or blurred imagery while correcting IMU drift [5]. IMUs are compact, efficient, and cost-effective, suiting lightweight AUV applications. Data-driven models replace hand-crafted pipelines; PoseLSTM [6] couples CNNs with LSTMs for motion blur and illumination robustness, while RNNs capture inertial dynamics. Advanced fusion strategies address noise and synchronization issues, e.g., selective fusion [7] and SelfVIO [8], which estimates six-DoF pose and depth from monocular images and IMU data without predefined sensor calibration. Policy-based adaptive VIO [9], [10] further reduces visual redundancy. Despite progress, large parameter counts and high computation cost still restrict deployment on underwater robots. The main limitations of existing VIO methods are:

1) *High computational cost:* Deep VIO models are resource-intensive, limiting real-time use on low-power underwater robots.
2) *Weak generalization:* Existing methods degrade under poor illumination, turbidity, or weak texture leading to reduced positioning accuracy.
3) *Dense data dependency:* Processing dense image and IMU data causes inefficiency; introducing sparsity data processing can improve VIO efficiency.

To address these challenges, we propose NeuroVIO, a hybrid end-to-end VIO framework integrating CNNs with spiking neural networks (SNNs). NeuroVIO combines robust CNN-based visual feature extraction with energy-efficient SNN-based inertial encoding. Visual features are converted into sparse spike sequences via adaptive leaky-integrate-and-fire (ALIF) neurons, while IMU data is encoded by a parallel SNN. A spike-LSTM fuses these multimodal temporal features, followed by an SNN regression head for relative pose estimation. The key contributions are:

- *End-to-End Hybrid Architecture:* Developed an end-to-end trainable CNN–SNN model for underwater VIO. The architecture leverages CNN-based visual feature extraction and SNN-based inertial processing, to capitalize the strengths of both paradigms for underwater VIO.
- *Direct Training of the Hybrid Model:* Enables direct optimization by training the hybrid model as a unified framework, simplifying the pipeline.
- *Adaptive Spike Representation:* Adaptive LIF (ALIF) neurons convert the continuous feature maps at each layer into sparse spike representations, reducing computational complexity.
- *Energy-Efficient Operation:* Validated on the open-source AQUALOC dataset, NeuroVIO demonstrates potential for energy-efficient long-term operation in challenging underwater environments

To the best of our knowledge, no prior work applies a hybrid CNN–SNN framework for visual–inertial pose estimation in underwater robots. Existing SNN-based research focuses on optical flow estimation from event cameras (event-only or event+RGB) without IMU integration for six-DoF pose estimation [11], [12], [13]. NeuroVIO is the first to unify CNN-based visual feature extraction, SNN-based inertial encoding, and SNN pose regression in an end-to-end model. This approach combines the high representational capacity of CNN-based visual feature extraction with SNNs for energy-efficient sequential inertial processing, enabling deployment on resource-constrained platforms.

This work is supported by Khalifa University under Awards No. RC1-2018-KUCARS-8474000136.

V. Sudevan and J. Dias are with Computer and Information Engineering, Khalifa University, Abu Dhabi, UAE [vidya.sudevan,jorge.dias]@ku.ac.ae

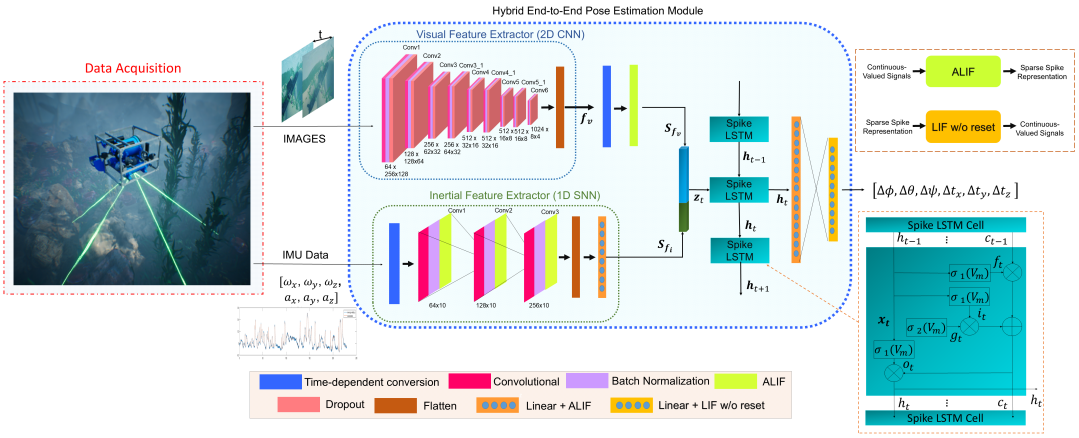


Fig. 1: Schematic representation of hybrid CNN-SNN framework for 6D pose estimation

II. SNN-BASED MULTIMODAL VISUAL-INERTIAL POSE ESTIMATION

The NeuroVIO framework presented in Fig. 1 integrates CNN-based visual feature extraction with SNN-based inertial processing. Visual features are converted into spike trains via adaptive LIF neurons, while raw inertial signals are first passed through the spike-coding module, which converts them into their corresponding sparse, spike-based representations. The spike-based multimodal features are fused and passed to a spike-LSTM, followed by a fully connected SNN regression head for 6-DoF pose estimation.

A. Spike Coding of Continuous-Valued Signals

Continuous signals are encoded with an adaptive threshold LIF mechanism. For visual inputs, 2D convolutions generate feature maps, where spikes fire when membrane potential (V) exceeds adaptive threshold membrane potential (V_{th}). Inertial data use 1D convolutions with the same mechanism. The mathematical formulation of the spike coding process is represented as:

1) Visual modality signals

$$S_{ij}^{k} = \Theta\Big[\sum_{c=1}^{C}\sum_{h=1}^{F_h}\sum_{w=1}^{F_w} x_{(i+h-1)(j+w-1)c} \cdot w_{hw}^{k} + b^{k} - V_{th}^{k}(t)\Big] \quad (1)$$

where S_{ij}^{k} is the output spike at $(i,\ j)$ in k^{th} output channel, $x_{(i+h-1)(j+w-1)c}$ is the input tensor at $(i+h-1,\ j+w-1)$, w_{hw}^{k} is the kernel weights of size F_h x F_w, b^{k} is the bias, and $V_{th}^{k}(t)$ is the time-dependent adaptive threshold of the spiking neuron.

2) Inertial signals

$$S_{i}^{k} = \Theta\Big[\sum_{c=1}^{C}\sum_{w=1}^{F_w} x_{(i+w-1)c} \cdot w_{wc}^{k} + b^{k} - V_{th}^{k}(t)\Big] \quad (2)$$

where S_{i}^{k} is the output spike at position i in k^{th} output channel, $x_{(i+w-1)c}$ is the inertial input signal at position $(i+w-1)$, w_{wc}^{k} is the kernel weights of size F_w x C.

B. Visual Feature Extractor

The visual feature extractor ($E_{f_v}(.)$) uses pre-trained FlowNetSimple [14], a 9-layer CNN-based architecture originally designed for optical flow, to extract geometric features from successive frames I_t and I_{t+1}. With stride-2 in the first six layers and kernel sizes reducing from 7×7 to 3×3 followed by Leaky-ReLU, features are flattened and passed through a fully connected layer to obtain $\boldsymbol{f_v}$:

$$\boldsymbol{f_v} = E_{f_v}(I_t, I_{t+1}) \in \mathcal{R}^{P_{seq} \times P_{f_v}}. \quad (3)$$

where I_t and I_{t+1} represent consecutive image frames captured at time t and $t+1$. These features are converted into time-dependent spike sequences ($\boldsymbol{f_v}$) via an ALIF layer:

$$\boldsymbol{S_{f_v}} = E_{f_v}(\boldsymbol{f_v}) \in \{0,1\}^{T \times P_{seq} \times P_{f_v}}. \quad (4)$$

C. Inertial Feature Extractor

The SNN-based inertial feature encoder ($E_{f_i}(.)$) consists of three 1D CNN layers and a fully connected layer, each followed by ALIF neurons. IMU measurements between two consecutive image frames (I_t and I_{t+1}) are aggregated and encoded into spikes $\boldsymbol{S_{f_i}}$:

$$\boldsymbol{S_{f_i}} = E_{f_i}(\boldsymbol{X_{t,imu}}, ..., \boldsymbol{X_{t+1,imu}}) \in \{0,1\}^{T \times P_{seq} \times P_{f_i}}. \quad (5)$$

where $X_{t,imu}$ and $X_{t+1,imu}$ represent the IMU measurements at time t and $t+1$.

D. Pose Regression

Spike-based visual and inertial features are concatenated and processed by a spike-LSTM [15], followed by linear layers with ALIF neurons, and a final LIF neuron layer without spike firing (LIF_{final}). The spike-LSTM extends conventional LSTMs by integrating spiking gates controlled by membrane potential V_m. Spike activations $\sigma_1(V_m)$ and $\sigma_2(V_m)$ regulate gate operations, producing spike or null outputs. As in standard LSTMs, the cell state c_t manages information flow: the forget gate f_t discards, the input gate i_t admits, and the auxiliary layer g_t is modulated by spike activation $\sigma_2(V_m)$. The final output depends on the output gate o_t and the cell state c_t [16]. The membrane potential at the last timestep of each neuron at the LIF_{final} layer is

the six-dimensional network-predicted pose $\hat{\boldsymbol{\mathcal{P}}} = \left[\hat{\boldsymbol{\mathcal{O}}}, \hat{\boldsymbol{\mathcal{T}}}\right] = \left[\hat{\phi}, \hat{\theta}, \hat{\psi}, \hat{t}_x, \hat{t}_y, \hat{t}_z\right] \in \mathcal{R}^6$.

$$\hat{\boldsymbol{\mathcal{P}}} = LIF_{final}(lin(ALIF(lin(E_{temp}(concat(\boldsymbol{S_{f_v}}, \boldsymbol{S_{f_i}})))))). \quad (6)$$

where $E_{temp}(.)$ is the temporal feature extractor module using spike-LSTM.

E. Loss Function

In this work, a unified loss function is used by integrating translational and rotational errors. For the network-predicted vectors $(\hat{\boldsymbol{\mathcal{T}}}, \hat{\boldsymbol{\mathcal{O}}})$ and their corresponding reference vectors $(\boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{O}})$, the loss is defined as:

$$\mathcal{L}_{pose} = \frac{1}{P_{seq}} \sum_{i=1}^{P_{seq}} \left(\left\| \hat{\boldsymbol{\mathcal{T}}}_{\mathrm{i}} - \boldsymbol{\mathcal{T}}_{\mathrm{i}} \right\|_2^2 + \gamma \left\| \hat{\boldsymbol{\mathcal{O}}}_{\mathrm{i}} - \boldsymbol{\mathcal{O}}_{\mathrm{i}} \right\|_2^2 \right) \quad (7)$$

where P_{seq} denotes the sequence length.

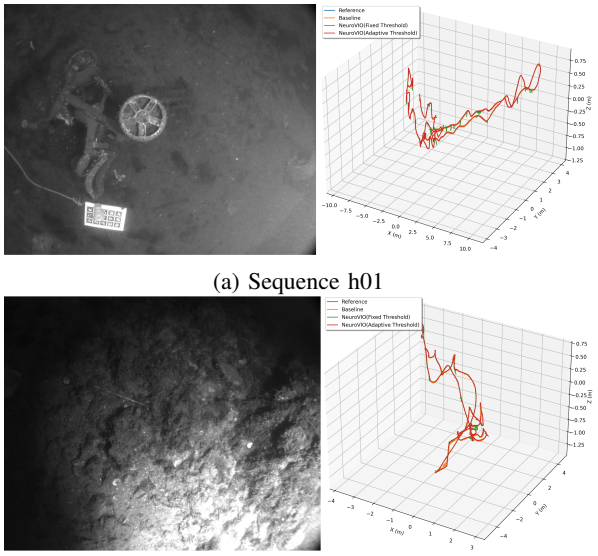
III. RESULTS

The NeuroVIO framework is implemented in three stages using SpikingJelly [17] and PyTorch, trained on an NVIDIA RTX 4070 GPU (16GB RAM) for 50 epochs with early stopping. The Adam optimizer [18] is used with a learning rate of 0.001 to ensure stable convergence and efficient gradient updates. Input images are resized to 512×256, timestep $T = 4$, and BPTT with surrogate gradients [19] is applied to update the network weights by minimizing the unified loss function $\mathcal{L}_{pose}$. The AQUALOC dataset [20] with linearly interpolated translational pose vectors and spherical linear interpolated [21] rotational vectors is used for training, validation, and testing of the NeuroVIO model. The AQUALOC-*Harbor site* was split into training $(\{h02, h04, h06\})$, validation $(\{h03, h05\})$, and testing $(\{h01, h07\})$. In our implementation, we assume that all images, IMU data, and corresponding pose vectors are temporally synchronized, with no missing values.

TABLE I: Hardware-oriented performance evaluation of baseline and NeuroVIO models.

Model	**# Operators** (G)	**Energy** (J)	ΔE (%)
Baseline	7.767	0.035	
NeuroVIO ($V_{th} = -0.025$)	7.788	0.007	80.38
NeuroVIO (Adaptive V_{th}**)**	7.781	0.007	80.40

In *Stage 1*, a CNN-based baseline model was designed for underwater environments with limited visual cues. It predicts relative translation and rotation from visual–inertial inputs. The visual encoder was initialized with pre-trained weights from a larger optical flow estimation dataset (FlyingChairs) [27], to leverage strong feature extraction capabilities. The IMU encoder used three 1D convolution layers, while temporal modeling employed LSTMs without dropout, ensuring stable training. This stage verified synchronization and feature extraction before spiking conversion. In *Stage 2*, the baseline was converted into a hybrid CNN-SNN: the pre-trained visual encoder was frozen, while inertial, temporal, and pose modules were implemented using spiking LIF neurons with fixed threshold membrane potential. This preserved visual features while introducing spike-driven processing, reduces computational complexity without significantly reducing pose estimation accuracy. *Stage 3* further optimizes the NeuroVIO model by replacing fixed LIF neurons with adaptive LIF (ALIF) neurons that learn threshold membrane potentials. This adaptation improves temporal coding and yields the final NeuroVIO framework, which balances CNN feature extraction with SNN energy efficiency.



(a) Sequence h01

(b) Sequence h07

Fig. 2: Reference trajectory and the trajectory sequences predicted by the baseline and NeuroVIO model with fixed and adaptive threshold membrane potential (a) h01 sequence, and (b) h07 sequence.

By advancing through three stages, our approach integrates analytical insights into a robust learning-based framework. The final NeuroVIO model fuses visual and inertial data, achieving energy efficiency while preserving accuracy in challenging environments. Experiments focused on energy efficiency, computational complexity, and pose estimation accuracy at each developmental stages. In *Stage 1*, the fully CNN-based baseline predicted 6D pose accurately but consumed significant energy. Replacing non-visual modules with spiking counterparts in *Stage 2* reduces energy from $0.035J$ to 0.007 J (-80.38%) with negligible change in operations (7.767 *GFLOPs* to 7.788 *GSOPs*). Using adaptive thresholds in *Stage 3*, the number of operations marginally decreases to 7.781 *GSOPs*, maintaining same energy cost. Adaptive NeuroVIO variant offers best efficiency without additional hardware overhead (refer to Table I).

On sequence h01 (refer to Table II), the baseline gave the lowest translational errors (ATE= 0.0174 m), while the adaptive SNN achieved comparable translation (0.0201 m) and superior rotation (ATE = 0.8167° vs. 2.1692°). On sequence h07, baseline again led in translation (0.0220 m), but the adaptive variant reduced rotation errors (1.3024° vs. 2.1366°) and attained the lowest drift (0.9154 m). Trajectory plots

TABLE II: Evaluation of translational (*trans*) and rotational (*rot*) estimates for baseline and NeuroVIO models with fixed and adaptive thresholds on sequences *h01* and *h07* from the *Harbor site* subset. Best values are shown in **bold**, second-best are underlined.

Sequence	Model	ATE		RPE		MAE		Drift Rate
		trans(m)	*rot*(°)	*trans*(m)	*rot*(°)	*trans*(m)	*rot*(°)	(m)
h01	**Baseline**	**0.0174**	2.1692	**0.0083**	1.2302	**0.0136**	1.5105	**0.7077**
	NeuroVIO ($V_{th} = -0.025$)	0.0224	<u>1.1296</u>	0.0109	<u>0.5845</u>	0.0182	<u>0.9092</u>	0.7690
	NeuroVIO (Adaptive V_{th}**)**	<u>0.0201</u>	**0.8167**	<u>0.0084</u>	**0.3524**	<u>0.0164</u>	**0.6668**	<u>0.7163</u>
h07	**Baseline**	**0.0220**	2.1366	**0.0101**	1.2565	**0.0175**	1.5143	<u>0.9475</u>
	NeuroVIO ($V_{th} = -0.025$)	0.0346	<u>1.5684</u>	0.0161	<u>0.8042</u>	0.0281	<u>1.2652</u>	1.2325
	NeuroVIO (Adaptive V_{th}**)**	<u>0.0304</u>	**1.3024**	<u>0.0126</u>	**0.5910**	<u>0.0248</u>	**1.0632**	**0.9154**

TABLE III: Effect of threshold membrane potential on VIO performance, evaluated on sequence *h01* with $V_{th} \in \{-0.075, -0.050, -0.025, 0.025, 0.050, 0.075\}$

V_{th}	ATE		RPE		MAE		Drift Rate
	trans(m)	*rot*(°)	*trans*(m)	*rot*(°)	*trans*(m)	*rot*(°)	(m)
-0.075	0.0184	0.8159	0.0076	0.3492	0.0150	0.6660	0.6626
-0.050	0.0184	0.8192	0.0076	0.3509	0.0150	0.6690	0.6624
-0.025	0.0224	1.1296	0.0109	0.5845	0.0182	0.9092	0.7690
0.025	0.2573	25.1098	0.1479	12.0338	0.2135	20.4953	10.3626
0.050	0.4211	63.3624	0.2346	28.4777	0.3496	51.7500	16.8742
0.075	0.5287	90.3052	0.2868	40.2785	0.4382	73.6282	20.6651

TABLE IV: Evaluation of NeuroVIO with SOTA VIO methods on the AQUALOC dataset using the $\mathcal{T}_{rmse}$ metric. Best values are in **bold**, second-best are underlined.

Methods	$h01$ **sequence** (m)	$h07$ **sequence** (m)
Geometry-based		
OKVIS [22]	0.0404	0.1171
ORB-SLAM3 [23]	0.0198	0.0212
CNN-based		
VINet [24]	0.0487	0.1495
U-VIO [25]	0.0570	0.0759
DU-VIO [26]	0.0111	0.0188
Baseline	**0.0083**	**0.0101**
SNN-based		
NeuroVIO ($V_{th} = -0.025$)	0.0109	0.0161
NeuroVIO (Adaptive V_{th})	<u>0.0084</u>	<u>0.0126</u>

(refer to Fig. 2) illustrate closer reference path alignment for the adaptive model, particularly in high-curvature regions.

A. Effect of Threshold Membrane Potential

Table III demonstrates that VIO performance is highly sensitive to the choice of fixed threshold membrane potential. Low values ($V_{th} = \{-0.075, -0.050\}$) yield acceptable accuracy, but higher thresholds cause sharp degradation. For instance, at $V_{th} = 0.075$, translational ATE increases to 0.5287 *m*, angular ATE to 90.3052°, and drift rate exceeds 20 *m*. This highlights the strong dependence of spiking neuron behavior on threshold selection. Careful manual tuning of V_{th} is required to ensure accuracy, which is labor-intensive and sequence- or task-dependent, limiting real-world practicality. In contrast, the adaptive NeuroVIO model learns optimal thresholds during training, adapting to data characteristics and eliminating the need for heuristic tuning.

B. Performance Comparison of NeuroVIO with state-of-the-art Methods

Table IV compares NeuroVIO with SOTA geometric and learning-based underwater VIO methods on AQUALOC sequences h01 and h07 using the $\mathcal{T}_{rmse}$ metric. The CNN-based Baseline achieves the lowest errors (0.0083 *m* and 0.0101 *m*, respectively), while the NeuroVIO variant with adaptive-threshold NeuroVIO ranks second-best (0.0084 *m* on *h01* and 0.0126 *m* on *h07*), closely matching it. The fixed-threshold NeuroVIO variant also demonstrates competitive accuracy (0.0109 *m* and 0.0161 *m*), outperforming state-of-the-art geometry-based methods (OKVIS, ORB-SLAM3) and CNN-based VINet and U-VIO, while being comparable to DU-VIO. These findings show that NeuroVIO matches baseline performance while drastically reducing energy. The adaptive variant, though slightly less accurate than the CNN baseline, closely approaches it and consistently outperforms most geometric and CNN-based methods.

Hardware analysis (refer to Table I) confirms the efficiency of spiking implementations. Both fixed and adaptive threshold NeuroVIO reduce energy from 0.035*J* to 0.007 *J* (-80.4%) with negligible change in number of operations. While the fixed-threshold model offers limited accuracy gains, it achieves substantial energy savings. The adaptive variant preserves these savings while delivering the best accuracy-per-joule ratio, making it well-suited for battery-constrained AUVs.

IV. CONCLUSIONS

This article introduced NeuroVIO, a hybrid CNN–SNN framework for underwater visual–inertial odometry. By combining time-encoded visual features with SNN-based inertial processing, NeuroVIO reduces energy consumption by 80.4% compared to its non-spiking counterpart while maintaining pose accuracy. This efficiency makes it a strong candidate for low-power neuromorphic and edge-AI deployment on energy-constrained underwater robots. Future work includes replacing the CNN visual encoder with an SNN-based module and integrating spike-driven image enhancement to evaluate performance under turbidity and distortion. In addition, NeuroVIO will be deployed on embedded hardware such as AMD Kria KV260 and Intel Loihi to validate real-time performance and energy efficiency. These steps will further demonstrate the potential of SNN-based pose estimation for autonomous marine robotics at the edge.

REFERENCES

[1] O. Álvarez-Tuñón, Y. Brodskiy, and E. Kayacan, "Monocular visual simultaneous localization and mapping:(r) evolution from geometry to deep learning-based pipelines," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 5, pp. 1990–2010, 2023.

[2] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE international conference on robotics and automation*, pp. 3565–3572, IEEE, 2007.

[3] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[4] B. Teixeira, H. Silva, A. Matos, and E. Silva, "Deep learning approaches assessment for underwater scene understanding and egomotion estimation," in *OCEANS 2019 MTS/IEEE SEATTLE*, pp. 1–9, IEEE, 2019.

[5] L. Jinyu, Y. Bangbang, C. Danpeng, W. Nan, Z. Guofeng, and B. Hujun, "Survey and evaluation of monocular visual-inertial slam algorithms for augmented reality," *Virtual Reality & Intelligent Hardware*, vol. 1, no. 4, pp. 386–410, 2019.

[6] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 627–637, 2017.

[7] C. Chen, S. Rosa, Y. Miao, C. X. Lu, W. Wu, A. Markham, and N. Trigoni, "Selective sensor fusion for neural visual-inertial odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10542–10551, 2019.

[8] Y. Almalioglu, M. Turan, M. R. U. Saputra, P. P. de Gusmão, A. Markham, and N. Trigoni, "Selfvio: Self-supervised deep monocular visual–inertial odometry and depth estimation," *Neural Networks*, vol. 150, pp. 119–136, 2022.

[9] M. Yang, Y. Chen, and H.-S. Kim, "Efficient deep visual and inertial odometry with adaptive visual modality selection," in *European Conference on Computer Vision*, pp. 233–250, Springer, 2022.

[10] Y. Lu, X. Yin, F. Qin, K. Huang, M. Zhang, and W. Huang, "A lightweight sensor fusion for neural visual inertial odometry," in *International Conference on Neural Computing for Advanced Applications*, pp. 46–59, Springer, 2023.

[11] R. Guamán-Rivera, J. Delpiano, and R. Verschae, "Event-based optical flow: Method categorisation and review of techniques that leverage deep learning," *Neurocomputing*, vol. 635, p. 129899, 2025.

[12] J. Courtois, B. Miramond, and A. Pegatoquet, "Spiking monocular event based 6d pose estimation for space application," *arXiv preprint arXiv:2501.02916*, 2025.

[13] H. Sun, J. Wang, W. Cai, D. Chen, Q. Liao, J. He, Y. Cui, D. Yao, and D. Guo, "St-flownet: An efficient spiking neural network for event-based optical flow estimation," *Neural Networks*, p. 107730, 2025.

[14] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2758–2766, 2015.

[15] A. Lotfi Rezaabad and S. Vishwanath, "Long short-term memory spiking networks and their applications," in *International Conference on Neuromorphic Systems 2020*, pp. 1–9, 2020.

[16] V. Sudevan, F. Zayer, S. Javed, H. Karki, G. De Masi, and J. Dias, "Hybrid-neuromorphic approach for underwater robotics applications: A conceptual framework," *arXiv preprint arXiv:2411.13962*, 2024.

[17] W. Fang, Y. Chen, J. Ding, Z. Yu, T. Masquelier, D. Chen, L. Huang, H. Zhou, G. Li, and Y. Tian, "Spikingjelly: An open-source machine learning infrastructure platform for spike-based intelligence," *Science Advances*, vol. 9, no. 40, p. eadi1480, 2023.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training spiking neural networks using lessons from deep learning," *Proceedings of the IEEE*, 2023.

[20] M. Ferrera, V. Creuze, J. Moras, and P. Trouvé-Peloux, "Aqualoc: An underwater dataset for visual–inertial–pressure localization," *The International Journal of Robotics Research*, vol. 38, no. 14, pp. 1549–1559, 2019.

[21] K. Shoemake, "Animating rotation with quaternion curves," in *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pp. 245–254, 1985.

[22] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[23] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE transactions on robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[24] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, 2017.

[25] V. Sudevan, F. Zayer, S. Javed, H. Karki, G. De Masi, and J. Dias, "Evaluating visual-selective visual-inertial odometry: An end-to-end multi-modal pose estimation framework for underwater environments," in *2023 21st International Conference on Advanced Robotics (ICAR)*, pp. 639–644, IEEE, 2023.

[26] V. Sudevan, F. Zayer, T. Hassan, S. Javed, H. Karki, G. De Masi, and J. Dias, "Dehazing-aided multi-rate multi-modal pose estimation framework for mitigating visual disturbances in extreme underwater domain," *arXiv preprint arXiv:2411.13988*, 2024.

[27] E. Ilg, T. Saikia, M. Keuper, and T. Brox, "Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation," in *European Conference on Computer Vision (ECCV)*, 2018.