Implementation of an efficient and optical-flow-based algorithm of depth estimation on autonomous nano quadcopters for obstacle avoidance

Chen-Fu Yeh
Institute of Systems Neuroscience
National Tsing Hua University
Hsinchu, Taiwan
chen_fu_yeh@lolab-nthu.org

Jia-Jun Lai
Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan
john04265@gmail.com

Sheng-Qian Li
Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan
shengqianli57@gmail.com

Fang-Kai Hsiao
Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan
zeus950068@gmail.com

Kuo-Chih Yu
Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan
bf1999416@gmail.com

Jie-Min Jhang
Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan
coo122567@gmail.com

Chung-Chuan Lo
Institute of Systems Neuroscience
National Tsing Hua University
Hsinchu, Taiwan
cclo@mx.nthu.edu.tw

Ya-Tang Yang*
Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan
ytyang@ee.nthu.edu.tw
*Corresponding author

***Abstract*—Nano quadcopters are small, agile, and cheap platforms well suited for deployment in narrow, cluttered environments. Due to their limited payload, these vehicles are highly constrained in computational power, making conventional vision-based navigation methods impractical for implementation. In this work, we present FlowDep, an efficient and optical flow-based algorithm for depth estimation. We draw inspiration from the low-resolution but efficient motion-detection mechanisms in insects. We successfully demonstrate the capabilities of the FlowDep by deploying it on a Bitcraze Crazyflie, a ~30 g nano quadcopter for obstacle avoidance with a single monocular camera. Additionally, we demonstrate the feasibility of the FlowDep algorithm in Gazebo simulation for obstacle avoidance in indoor and outdoor test environments.**

***Keywords*—depth estimation, obstacle avoidance, optical flow, micro aerial vehicle, nano quadcopters**

I. Introduction

Safe and reliable navigation of autonomous aerial systems in narrow, cluttered, GPS-denied, and unknown environments is one of the main open challenges in the field of robotics. Because of their small size and agility, micro air vehicles (MAVs) are optimal for this task [1], [2]. Nano quadcopters are a variety of MAVs that are characterized by minimal weight (typically below the range of ~100 g) and size (typically with rotor to rotor distance of 10 cm). Despite of small size, these nano quadcopters have shown impressive performance on tasks such as exploration [3] and gas source seeking [4].

The traditional approach of passive depth estimation is based on stereo vision, which requires two precisely calibrated cameras, and the depth information is calculated based on disparity. However, this method is limited by the higher cost (two cameras) and the physical separation between the two cameras. On a nano quadcopter, the rotor-to-rotor distance is at most on the order of ~ 10 cm and hence offers a maximal distance range of estimation to a few meters for stereo vision. In this work, we turn our attention to optical flow, one of the most important monocular visual cues for navigation. Until now, it has been used for high payload capacity [5,6]. Also, a bio-inspired method [7-11]. Also, the field of monocular optical estimation has shifted toward deep learning [12-24], and there is limited work on obstacle avoidance based on optical flow implemented on nano quadcopters [25]. De Croon's group has developed NanoFlowNet based on real-time dense optical flow on a nano quadcopter using a lightweight convolution neural network. The neural network is trained with a dataset with known optical flow ground truth and real-time inference on the ultralow-power GAP8 multi-core microprocessor on the Bitcraze AI-deck. Although not based on optical flow, in related work, very impressive flight speed is achieved based on depth estimation is achieved based on a novel millimeter form factor 64 pixels multizone time-of-flight (ToF) sensor [26]. The autonomous nanosize drone reaches 100% reliability at 0.5 m/s in a generic and previously unexplored indoor environment.

Previously, we have reported both the theoretical details of the FlowDep algorithm and the implementation of FlowDep on a ground mini-vehicle based on Raspberry Pi 4 Model B with

The authors would like to acknowledge funding support from the National Science and Technology Council under grant number 113-2218-E-007-019 and National Tsing Hua University under grant number 113Q2703E1.

the low resolution (320 pixel x 240 pixel) image data from Arducam camera OV9782 as the computation platform [27]. It is not clear whether or not the idea of FlowDep can be implemented on aerial vehicles such as nano quadcopters. In this work, we present the implementation of FlowDep on a nano quadcopter and demonstrate its capability for obstacle avoidance. Additionally, as an independent validation, we also implement the FlowDep algorithm in a simulation environment in Gazebo to prove the general applicability of the FlowDep algorithm. It also allows us to freely vary the critical parameters such as flight velocity in the simulated flight condition without conducting potentially abusive experimentation on actual micro aerial vehicles and also relieves us from any noisy perturbation from sensors or environmental disturbance.

As a summary, the main processing step of the FlowDep algorithm is shown in Fig. 1. The algorithm assumes an infinite wall on the x-y plane in front of the camera and computes a "predicted" optical flow for every point on the wall based on the movement of the camera. The depth information of an observed object can be extracted by comparing the predicted and the observed optical flows. FlowDep takes the optical flow (computed using the Dense Inverse Search method [28]) and the motion signals of the micro aerial vehicle as the input. The motion signals contain rotation components from the onboard inertial measurement unit and translation components from the estimated horizontal velocity. The former is used to perform de-rotation for the observed optical flow, and the translation components are used to generate the predicted optical flow.

Similar to many structure-from-motion methods, FlowDep can operate as a keyframe-based algorithm [29] by selecting historical frames as keyframes to adjust the effective depth range. By leveraging the odometry provided by the onboard IMU and the downward Flow-deck, the system obtains the relative camera pose with an absolute scale. This information allows us to set criteria for selecting a previous frame for optical flow computation. One approach is to assume a relatively stable flying speed and reduce the camera's frame rate, effectively spacing out the frames used for depth estimation. Alternatively, we can select a frame based on a predefined absolute distance from the current frame. Both strategies enable FlowDep to dynamically adjust the depth range, ensuring more accurate depth estimation in varying operational conditions.

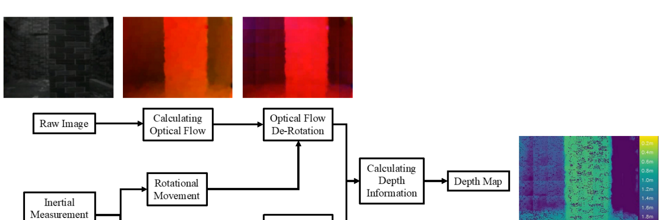

Fig. 1. The main processing steps of FlowDep.

II. Methods

A. Implementation of the algorithm on the nano quadcopter

We deploy the proposed FlowDep algorithm on a Crazyflie 2.x equipped with the AI-deck and the Flow-deck for the task of vision-based obstacle avoidance. We use the AI-deck to capture images with the front-facing camera and to run processing on laptop CPUs. The downward-facing optical Flow-deck is used to provide the velocity data in the FlowDep algorithm. (The Flow-deck is based on PMW3901, an optical flow ASIC that computes the flow internally and provides a difference in pixels between each frame.) To obtain reliable estimation of the flight velocity Vx and Vy, we have placed a well-designed texture carpet composed of the scene of forests from the satellite images of the central park in New York. The total flight platform weighs in at 34 g. See Fig. 2 for a picture of the platform. The AI-deck of the Crazyflie provides image streaming via Wi-Fi and uses the implemented FlowDep algorithm laptop CPU for depth estimation and obstacle avoidance decision-making. The high-level flight control command is computed with the control strategy and sent back to the nano quadcopter. Fig. 2b provides a summary of the fusion of the image, inertial measurement unit, and position data in the FlowDep algorithm to give the depth estimation.

B. Test environments

We compose indoor environments for obstacle avoidance. A cluttered environment is constructed with obstacles such as textured poles placed inside. In Fig. 3a, a single rectangular pillar with a brick wall is placed. In Fig. 3b, in a representative environment, the textured poles are made of two cardboard cylinders wrapped with synthetic plants. The environment is enclosed with textured panels to provide the "predicted" optical flow of the background. The ground carpet consists of forest texture optimized for stable and smooth flight (Fig. 3c). The carpet also provides texture for the Flow-deck of the down-facing camera to measure horizontal velocity.

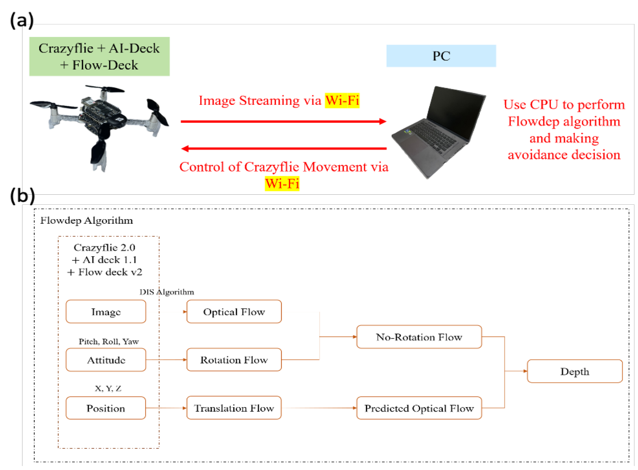

Fig. 2. Experiment setup (a) The AI-deck of the Crazyflie provides image streaming via Wi-Fi and uses the implemented FlowDep algorithm on the laptop CPU. (b) Summary of the fusion of image, inertial measurement unit, and position data in FlowDep algorithm to give the depth estimation.

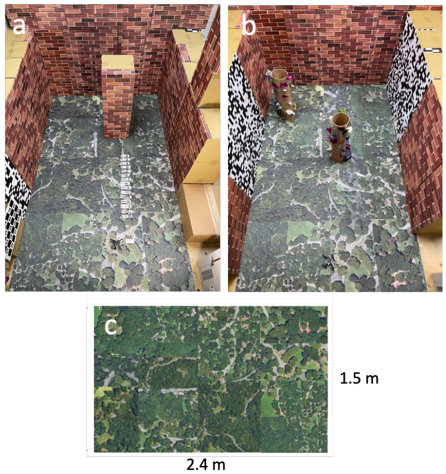

Fig. 3. Test environments (a) Single obstacle test environment with a rectangular pole. (b) Multiple obstacle test environment with two textured cylinders decorated with synthetic plants. (c) The carpet used on the ground of test environments.

C. *Control strategy for obstacle avoidance*

We adopt a simple control strategy tailored to micro aerial vehicles (MAVs). The core of our control strategy is from the reference[30]. The image obtained from a monocular camera is first split into two horizontal and vertical half-planes. The desired heading direction and pitch rate are then determined by comparing the sum of optical flows between half-planes horizontally and vertically, respectively. Through testing with both single and multiple obstacles, the FlowDep algorithm demonstrated strong performance in obstacle avoidance.

D. *Simulation in Gazebo environment*

The simulation is done on a laptop computer with Intel Core i9-12900HX and 16 GB memory. The computation is also accelerated with GPU (Nvidia GeForce RTX 4080). The Robot Operating System (ROS) is used as the operating system under Ubuntu 20.04.6 LTS. The simulation uses an open-source software called PX4 Software-In-The-Loop (SITL) as the flight control software [31]. We chose Bayland from Gazebo as the outdoor test environment (Fig. 4). This scene consists of arrays of trees with rich textures ideal for optical flow-based algorithms. We also constructed an indoor environment that closely mimics our actual flight environment for the nano quadcopter (Fig. 5). Before we simulate FlowDep, we carry out benchmark flights such as circular flights to make sure the inertial measurement unit (IMU) is output as a sanity check.

Fig. 4. Bayland outdoor environment in Gazebo

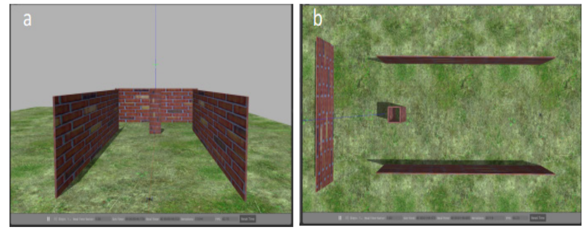

Fig. 5. Indoor environment in Gazebo

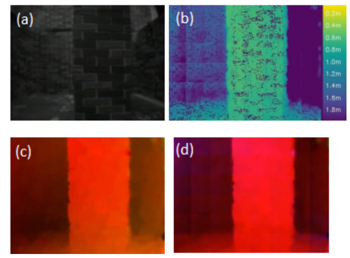

Fig. 6 (a) Raw image (b) Depth estimation (c) Optical flow (d) De-rotated optical flow

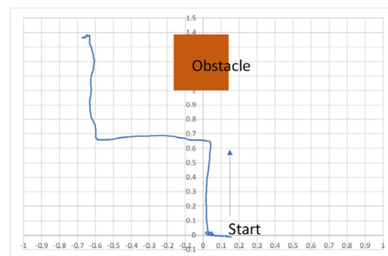

Fig. 7. Obstacle avoidance experiment. The unit of the coordinate is in meters.

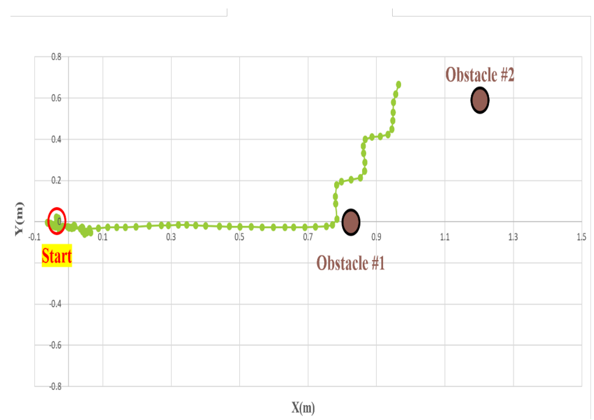

Fig. 8. Multiple obstacle avoidance experiment. The unit of the coordinate is in meters.

III. RESULTS

A. *Single and multiple obstacle avoidance of the nano quadcopter*

Fig. 6 shows the depth estimation from a greyscale input image and intermediate results of optical flow and de-rotated optical flow. Fig. 7 shows obstacle avoidance of a single obstacle of a square pillar with brick wallpaper as the texture. The blue curve is the flight trajectory extracted from video from an external web camera. The forward velocity is 0.15 m/s and the FlowDep runs at 7 fps on the laptop CPUs. Similarly, Fig. 8 shows the multiple obstacle avoidance at a forward velocity of 0.2 m/s.

We have currently explored the feasibility of implementing FlowDep directly on the Bitcraze AI-deck. Utilizing the GAP8 SDK—which supports the RISC-V GNU toolchain—we can cross-compile C code to run on the AI-deck. Our preliminary implementation, using a pure-C sparse forward-compositional Lucas-Kanade optical flow method, has achieved onboard processing speeds of approximately 6–7 FPS. Moreover, theoretical improvements—such as switching to an inverse-compositional Lucas-Kanade approach [32] and employing a less computationally intensive pixel interpolation method—could potentially accelerate the algorithm by up to 4×. Given that these findings are preliminary and further validation is necessary, we have outlined this direction as part of our future work.

B. *Obstacle avoidance on simulation on Gazebo*

The raw image from the simulation is used to calculate the optical flow and the simulated data from the inertial measurement unit is fused with image data as input of the FlowDep algorithm. Fig. 9 shows the representative depth estimation as compared to the depth data from the simulated stereo camera in Gazebo. The obstacles consist of an array of threes with rich textures. We choose the forward velocity as the critical parameter to vary but fix the control strategy. For each forward velocity, we slightly vary the incident angle to produce perturbed flight conditions. We conduct 10 trials for each forward velocity. The FlowDep algorithm works for a certain range of velocity even for multiple obstacle avoidance (Fig. 10). At low velocity v= 0.8 m/s, the drone runs into the forest and yet no collision occurs. For high-velocity flight v= 3.2 m/s, around 60 % of the trials result in collision. In general, higher forward flight velocity results in high collision probability is understandable as the drone does not have enough time to make inferences with a fixed control strategy. Improvement in collision probability is possible if we allow the control strategy to vary to adapt to high flight velocity. Similarly, the FlowDep algorithm works for obstacle avoidance in the indoor environment (Fig. 11).

C. *Robustness to Environmental Variability*

In our current implementation, FlowDep relies on accurate optical flow estimation, which can be challenging in environments with limited textures. As is well known, missing textures often lead to less reliable optical flow. To address this, we already employ a pyramidal implementation [33] of optical flow, which helps mitigate scale issues. For future work, we plan to further enhance robustness by incorporating additional contextual information and approach this problem via two promising avenues. First, spatial context via CNNs: Leveraging convolutional neural networks [25] could help the algorithm infer motion in low-texture areas by understanding the broader spatial context. Second, temporal context via Bayesian Filtering: Incorporating Bayesian filtering might not only refine the optical flow estimates over time but also aid in detecting dynamic obstacles. However, further investigation is required to fully assess its benefits within the FlowDep framework.

IV. Conclusion

In this work, we introduced a lightweight architecture to implement FlowDep for depth estimation. We implemented FlowDep in a real-world obstacle avoidance application a Bitcraze Crazyflie nano quadcopter. In simulation, FlowDep can assist drones in maintaining stable flight and avoiding collisions with obstacles in both complex outdoor terrains like forests and indoor environments. The application of the FlowDep algorithm demonstrates the immense potential of bio-inspired systems in robot navigation and also suggests a new direction for future low-cost robotic vision systems.

Fig. 9. Depth estimation from a representative Bayland scenery consists of multiple trees. (a) Raw image from the test environment (b) Depth estimation (c) Ground truth of the depth from the simulated stereo camera in Gazebo.

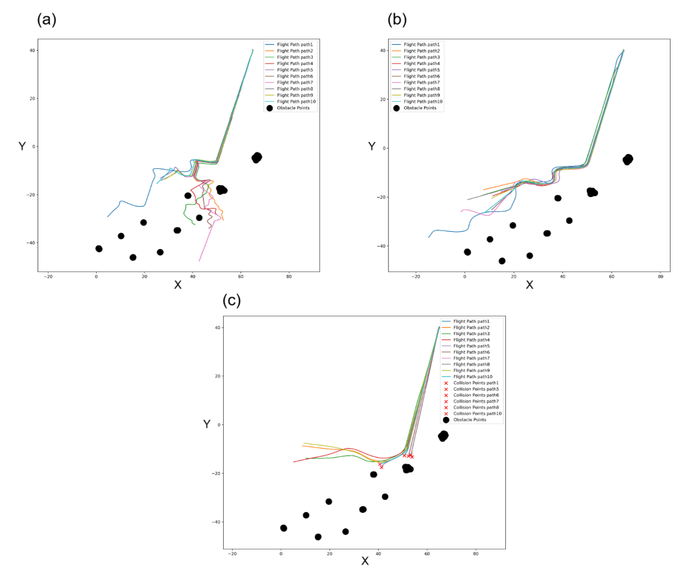

Fig. 10. The trajectory of multiple obstacle avoidance simulations at different forward velocities for the outdoor environment (a) 0.8 m/s (b) 1.6m/s (c) 3.2 m/s

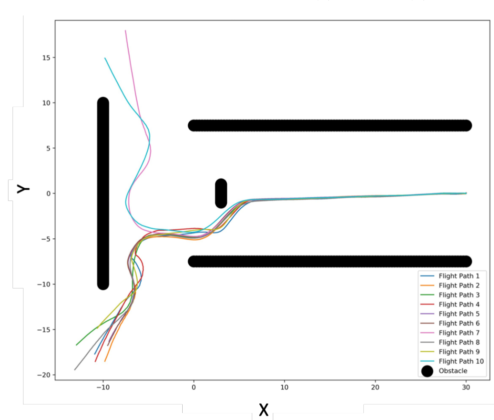

Fig. 11. Trajectory of drone for obstacle avoidance in indoor environment.

REFERENCES

[1] D. Floreano and R. J. Wood, "Science, technology and the future of small autonomous drones," Nature, vol. 521, no. 7553, pp. 460–466, may 2015.

[2] B. Bodin, H. Wagstaff, S. Saecdi, L. Nardi, E. Vespa, J. Mawer, A. Nisbet, M. Lujan, S. Furber, A. J. Davison, P. H. Kelly, and M. F. O'Boyle, "SLAMBench2: Multi-Objective Head-to-Head Benchmark- ing for Visual SLAM," in Proceedings - IEEE International Conference on Robotics and Automation, sep 2018, pp. 3637–3644.

[3] K. N. McGuire, C. de Wagter, K. Tuyls, H. J. Kappen, and G. C. H. E. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," Science Robotics, vol. 4, no. 35, oct 2019.

[4] B. P. Duisterhof, S. Li, J. Burgues, V. J. Reddi, and G. C. H. E. de Croon, "Sniffy Bug: A Fully Autonomous Swarm of Gas-Seeking Nano Quadcopters in Cluttered Environments," in IEEE International Conference on Intelligent Robots and Systems, jul 2021, pp. 9099–9106.

[5] P. Gao, D. Zhang, Q. Fang, and S. Jin, "Obstacle avoidance for micro quadrotor based on optical flow," in Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017, jul 2017, pp. 4033– 4037.

[6] N. J. Sanket, C. D. Singh, K. Ganguly, C. Fermuller, and Y. Aloimonos, "GapFlyt: Active vision based minimalist structure- less gap detection for quadrotor flight," IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 2799–2806, 2018.

[7] J. Conroy, G. Gremillion, B. Ranganathan, and J. S. Humbert, "Implementation of wide-field integration of optic flow for autonomous quadrotor navigation," in *Autonomous Robots*, vol. 27, no. 3, oct 2009, pp. 189–198.

[8] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart, "MAV navigation through indoor corridors using optical flow," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2010, pp. 3361–3368.

[9] G. C. H. E. de Croon, "Monocular distance estimation with optical flow maneuvers and efference copies: A stability-based strategy," *Bioinspiration and Biomimetics*, vol. 11, no. 1, jan 2016.

[10] J. R. Serres and F. Ruffier, "Optic flow-based collision-free strategies: From insects to robots," *Arthropod Structure and Development*, vol. 46, no. 5, pp. 703–717, sep 2017.

[11] G. C. H. E. de Croon, C. De Wagter, and T. Seidl, "Enhancing optical-flow-based control by learning visual appearance cues for flying robots," *Nature Machine Intelligence*, vol. 3, no. 1, pp. 33–41, jan 2021.

[12] A.Dosovitskiy, P.Fischery, E.Ilg, P.Hausser, C.Hazirbas, V.Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.

[13] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition*, nov 2017, pp. 2720–2729.

[14] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1647–1655.

[15] S. Zhao, X. Li, and O. El Farouk Bourahla, "Deep optical flow estimation via multi-scale correspondence structure learning," in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 0, jul 2017, pp. 3490–3496.

[16] D. Sun, X. Yang, M. Y. Liu, and J. Kautz, "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2017, pp. 8934–8943.

[17] T. W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8981–8989.

[18] Z. Yin, T. Darrell, and F. Yu, "Hierarchical discrete distribution decomposition for match density estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, dec 2019, pp. 6037–6046.*

[19] G. Yang and D. Ramanan, "Volumetric correspondence networks for optical flow," in *Advances in Neural Information Processing Systems*,

[20] vol. 32, 2019.

[21] T. W. Hui, X. Tang, and C. C. Loy, "A Lightweight Optical Flow CNN - Revisiting Data Fidelity and Regularization," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 8, pp. 2555–2569, feb 2021.

[22] T. W. Hui and C. C. Loy, "LiteFlowNet3: Resolving Correspondence Ambiguity for More Accurate Optical Flow Estimation," in *European Conference on Computer Vision*, 2020, pp. 169–184.

[23] S. Zhao, Y. Sheng, Y. Dong, E. I. Chang, and Y. Xu, "Maskflownet: Asymmetric feature matching with learnable occlusion mask," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, mar 2020, pp. 6277–6286.

[24] Z. Teed and J. Deng, "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow (Extended Abstract)," in *European Conference on Computer Vision*, aug 2020, pp. 402–419.

[25] R. J. Bouwmeester, F. Paredes-Vallés and G. C. H. E. de Croon, "NanoFlowNet: Real-time Dense Optical Flow on a Nano Quadcopter," 2023 IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom, 2023, pp. 1996-2003, doi: 10.1109/ICRA48891.2023.10161258.

[26] H. Müller, V. Niculescu, T. Polonelli, M. Magno and L. Benini, "Robust and Efficient Depth-Based Obstacle Avoidance for Autonomous Miniaturized UAVs," in *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4935-4951, Dec. 2023, doi: 10.1109/TRO.2023.3315710.

[27] Chen-Fu Yeh, Chao-Yang Tang, Tsu-Chiao Chen, et al. FlowDep - An efficient and optical-flow-based algorithm of obstacle detection for autonomous mini-vehicles. *TechRxiv.*April 03, 2024.

[28] Till Kroeger et al. *Fast Optical Flow using Dense Inverse Search*. Mar. 11, 2016. DOI: 10.48550/arXiv.1603.03590. arXiv: 1603. 03590[cs]. URL: http://arxiv.org/abs/1603.03590

[29] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Nara, Japan, 2007, pp. 225–234. doi: 10.1109/ISMAR.2007.4538852.

[30] Cho, G.; Kim, J.; Oh, H. Vision-Based Obstacle Avoidance Strategies for MAVs Using Optical Flows in 3-D Textured Environments. *Sensors* **2019**, *19*, 2523.

[31] Lorenz Meier, Dominik Honegger, Marc Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," 2015 IEEE International Conference on Robotics and Automation (ICRA) Washington State Convention Center Seattle, Washington, May 26-30, 2015.

[32] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, pp. 221–255, 2004. doi: 10.1023/B:VISI.0000011205.11775.fd.

[33] J.-Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker: Description of the algorithm," Intel Corporation, Microprocessor Research Labs, 2001. [Online]. Available: http://robots.stanford.edu/cs223b04/algo_tracking.pdf