Person localization and distance determination using the raycast method

Goran Paulin
Kreativni odjel d.o.o.
Rijeka, Croatia
orcid.org/0000-0002-6885-5393

Sasa Sambolek
High school Tina Ujevića
Kutina, Croatia
orcid.org/0000-0002-5287-2041

Marina Ivasic-Kos
Department of Informatics
University of Rijeka
Rijeka, Croatia
orcid.org/0000-0002-1940-5089

***Abstract*—By using drones in search and rescue (SAR) missions, missing person detection is possible during or after the flight by analyzing the recorded material. However, person localization is equally important so that rescuers can approach the person in the shortest possible time. We propose a raycast method to determine both a person's location and the distance from the drone, using a sequence of monocular drone images. The proposed method has been tested in silico, using a custom-made procedural simulator, calibrated for windless and windy conditions. We concluded that multiple raycasting solves unreliable telemetry problems and that there is an optimal number of required iterations, depending on the telemetry noise of a specific drone.**

Keywords—raycasting, drone imagery, localization, distance determination, search and rescue missions

I. INTRODUCTION

Search and rescue (SAR) mission in the wild is a complex task due to the limited time required to operate and a large search area that increase over time from the received rescue call. Due to the large search area, all possible resources are activated (people, search dogs, vehicles, helicopters, IR cameras, drones...).

Detection of missing persons using drones is possible during the flight or after the flight by analyzing the recorded material [1]. In SAR missions, in addition to the person detection, the person localization and drone distance estimation are equally important so that rescuers can approach the person in the shortest possible time. If the detection occurred during a drone flight, the remote pilot could determine the person's position on the map and distance from the drone according to the drone's position using the metadata of drone photos such as the drone's GPS position, altitude, and orientation, as well as camera orientation (roll/pitch/yaw). In the case of offline detection, i.e., when analyzing the recorded material, the localization of the person is limited to the recorded material, which can be a single monocular RGB photo.

Distance estimation in computer vision is most commonly done via stereo vision, in which images from two stereo cameras are used to triangulate and estimate distances to objects and potential obstacles [2]. Xiaoming et al. proposed a real-time method that can measure distance using a lens radius, focal length, sensor height, distance from sensor center to lens center, the inclination of the sensor, and the number of sensor bars [3]. Object distance can be obtained by applying image distance to the lens formula.

In [4], using the camera's field of view (FOV) and camera height, they estimated the distance and GPS location of the detected object. In this case, as in [5], the camera looks perpendicular to the ground, and the distance of the camera focal length and the flight height of the UAV determine the scale of the image. By converting the image to the corresponding bird's-eye view using the Inverse Perspective Mapping (IPM) algorithm, it is possible to estimate distance [6]. Inverse Perspective Mapping (IPM) [7] is a mathematical technique for transforming a coordinate system from one perspective to another.

DistNet [8] is another approach, where authors used the object's bounding boxes resulting from the YOLO object classification, processed to calculate the features, bounding boxes' parameters. Based on the input features, the trained DistNet outputs the estimated distance of the object to the camera sensor.

The authors [9] introduced a base model to directly predict distances (in meters) from a given RGB image and object bounding boxes as the first attempt to utilize deep learning techniques for object-specific distance estimation. To predict a distance, they use the distance regressor, which contains three fully connected (FC) layers and applies a softplus activation function on the last fully connected layer.

In this paper, for the case of offline search of a missing person, it is proposed to use drone photo metadata and a raycast method to determine the person's location and distance from the drone. A raycast is a process of tracing geometric rays from

the camera, finding line-surface intersection points [10]. It was initially invented as the methodological basis for a CAD/CAM solid modeling system.

Fig. 1 – Aerial image, taken by a drone (left). Detection of persons (green bounding boxes) performed by YOLO v4 (right).

The rest of the paper is organized as follows: in Section II. a description of the proposed raycast method for person localization and distance determination is given. A custom-made procedural simulator used to test the reliability of the proposed method considering a telemetry noise is presented in Section III. and calibration of the simulator and discussion of error variations in Section IV. The paper ends with a conclusion and a proposal for future research.

II. Proposed Raycast method

We propose a raycast method to estimate the distance of the detected person from the drone position.

Prerequisites for using the method are:

- a sequence of aerial images, taken by a drone (Fig. 1, left)
- known specification of drone's camera optics
- drone and camera telemetry for each image
- performed detection of persons in each image, using a neural network (Fig. 1, right)
- normalized device coordinates (NDC) of the center of the bounding box of the detected person.

The 2D coordinates of the center of the object, or in our case the person, that we receive from the detector, i.e., YOLO, for each bounding box are first normalized, and then these coordinates are transformed into 3D coordinates taking into account the position and orientation of the camera. In our specific example, the center of the red blob (representing the person's coordinate) is at the (2D) NDC coordinates f (0.635268, 0.663839). Note: (0,0) is the lower left, and (1,1) is the upper right corner of the 2D image. For the transformation of 2D coordinates into 3D coordinates, we considered the camera coordinates, which were, i.e., $\boldsymbol{c}$ (85, 62, 80), and the orientation $\boldsymbol{o}$ (-30, 45, 0), so the position of the point f in the 3D space becomes $\boldsymbol{t}$ (73.7975, 53.3224, 65.6279).

The raycast direction, in the example the vector $\boldsymbol{r}$ (-0.555868, -0.42924, -0.711873), is then calculated as the normalized vector of the distance of the camera position $\boldsymbol{c}$ from the point $\boldsymbol{t}$. In the further procedure, the raycast algorithm looks for the intersection of a line running from the center of the camera in the direction of the raycast vector $\boldsymbol{r}$ and somewhere (potentially) intersecting the terrain. The digital 3D model of the terrain is used as a collider. The point where the terrain is being intersected is $\boldsymbol{p}$, and from it, we can read the geolocation (as the coordinates of the point $\boldsymbol{p}$). The resulting point $\boldsymbol{p}$ matches the center of the 2D bounding box of a detected person in the initial image (Fig 2).

Fig. 2 – Example of person detection with the corresponding geographic coordinates.

The distance d of the detected person is calculated as the distance between the camera position $\boldsymbol{c}$ and the determined 3D point $\boldsymbol{p}$ on the terrain (1).

$$d(\boldsymbol{c},\boldsymbol{p}) = \sqrt{(c_x - p_x)^2 + \left(c_y - p_y\right)^2 + (c_z - p_z)^2} \qquad (1)$$

III. Simulator

To test the method, we built a simulator and calibrated it by measuring drone data. In the domain of autonomous flying, simulators are most often used to build synthetic datasets [11], [12] used in supervised learning, but also to evaluate the performance of reinforced learning models with emphasis on their energy efficiency when used on specific hardware platforms [13]. In addition to annotated images, some simulators generate various aerial vehicle sensor data [14]. Simulator often relies on using a game engine [15] or a modified commercial computer game to conduct a specific simulation. Instead, we opted for a digital content creation tool that gives us easy access to data and powerful data visualization ability.

The proposed method has been tested in silico, using a custom-made procedural simulator. The simulation is carried out through two stages (1 and 2a-2e). In the first stage, a 3D scene is prepared, and in the second stage, telemetry noise is added to model the actual conditions of application of the raycast method, which involves various atmospheric influences and measurement errors that affect signal accuracy and includes unreliable telemetry.

The simulation stages are as follows:

1) 3D scene preparation: a terrain is generated, a person is positioned, a drone/camera is positioned and oriented, and the camera optics (lens width, aspect ratio) are adjusted. This setup represents perfect drone/camera telemetry, and it is used as ground truth (GT).

2a) Addition of telemetry noise: the modeled error variation is applied to the position and orientation of the drone/camera in each frame.

2b) Photo/video shooting is simulated at arbitrary resolution (in the example 160x160 px, aspect ratio 16:9), also using raycasting, where the number of rays corresponds to the number of target pixels. By varying the resolution, it is possible to test the detection tolerance, finding the minimum number of pixels required to represent a person.

2c) Detection is simulated without the use of neural networks: the center of a group of pixels is sought (a red blob representing a person in the example), resulting in NDC coordinates.

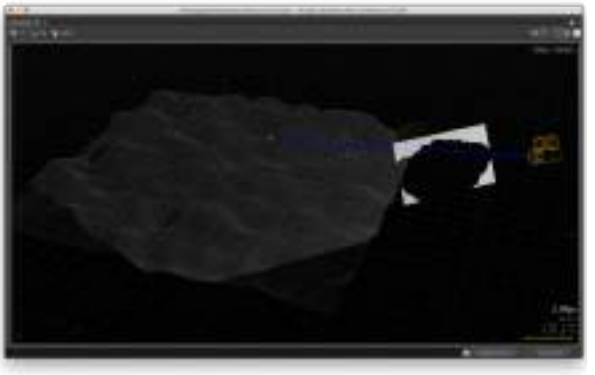

Fig. 3 shows the entire pixel matrix (160x160) projected onto the terrain, but in this step, we use only a single ray (projection of the NDC coordinates) which gives us the 3D coordinates of the person.

2d) Raycasting is performed (Fig. 3). The accuracy depends on 2 components: the resolution of the digital terrain (1 meter in the example) and the resolution of the drone image (160x160 px in the example).

2e) The error is calculated as the distance of the raycasted point from the person's GT position.

Iterating steps 2a-2e a desired number of times (240 in the example), the average error is calculated as the central coordinate of all previously obtained by raycasts.

The simulator allows generating the random terrain (patch 100x100m in the example). However, in a real application, the use of a digital elevation map (DEM), meshed LiDAR point cloud, or sculpted 3D model is expected. A prerequisite for

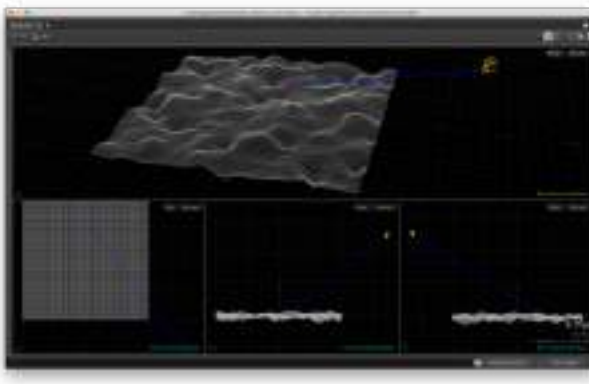

Fig. 4 shows the terrain (white) and the drone (yellow camera, with visible orientation) at the height of 62 m. The red sphere represents a person, but a 3D model of a person can be used instead (ragdoll models are ideal for representing casualties). In this case, the result (distance) is 152.93 m, visualized with the blue line (from camera to person).

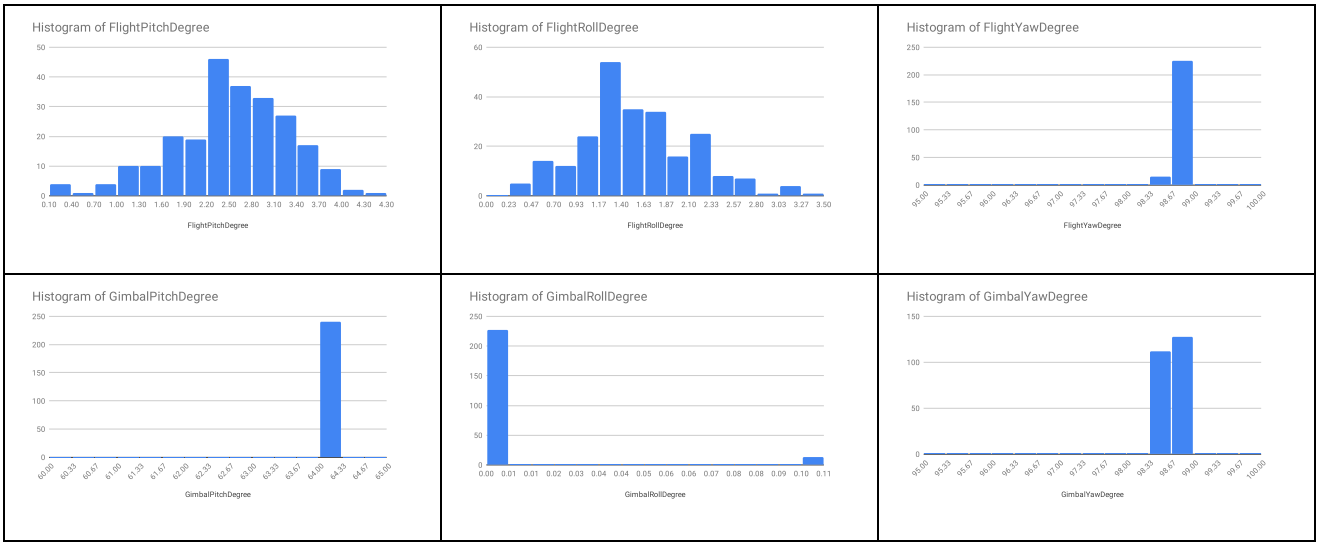

Fig. 2 – Windless conditions. Histograms of roll/pitch/yaw degree of drone (1st row) and camera (2nd row).

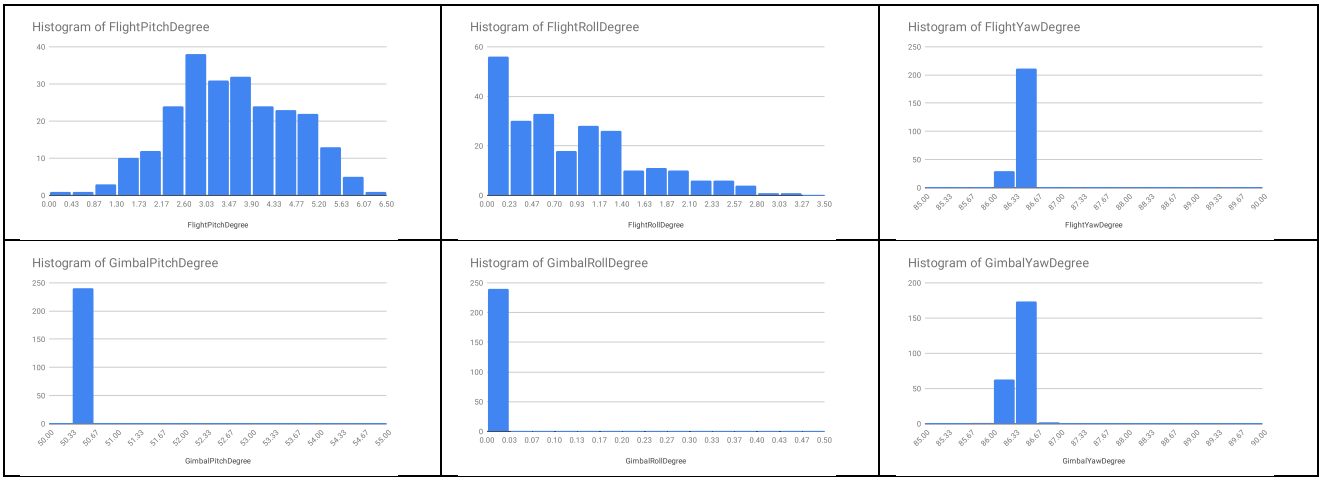

Fig. 3 – Gentle breeze (4 m/s). Histograms of roll/pitch/yaw degree of drone (1st row) and camera (2nd row).

using the terrain is that it can act as a collider. The position and shape of the person are determined by the user (a simple red sphere was used in the example, Fig. 4).

The outputs of the simulation are the estimated distance (in meters) and the geographic coordinates. The digital terrain within the simulation can contain specific data such as the difficulty of the terrain (different types of soil, minefields) and can automatically provide information about slopes (cliffs) which both can be used for pathfinding from the starting point to the person located by drone in search and rescue missions.

IV. Calibration

To calibrate the simulator and model the error variation, measurements of the drone Phantom 4 Advanced [16] were used with conditions of 4 m/s wind and without wind.

The drone hovers for 8 minutes at a height approximately corresponding to that from which it searches the terrain in real conditions [17]. The camera is aimed at -60 degrees relative to the horizon and captures 1 image every 2 seconds (for a total of 240 samples).

Histograms (Fig. 5 and Fig. 6) show a significant variation of drone's pitch (0.3-6.1 degrees with the

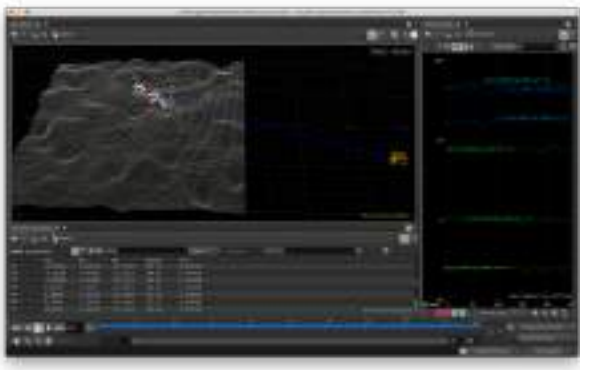

Fig. 1 shows replicated error distribution. The white dots represent 240 iterations of simulated unstable telemetry, and the red sphere the position of the person (GT). The shape of the noise function can be seen on the right.

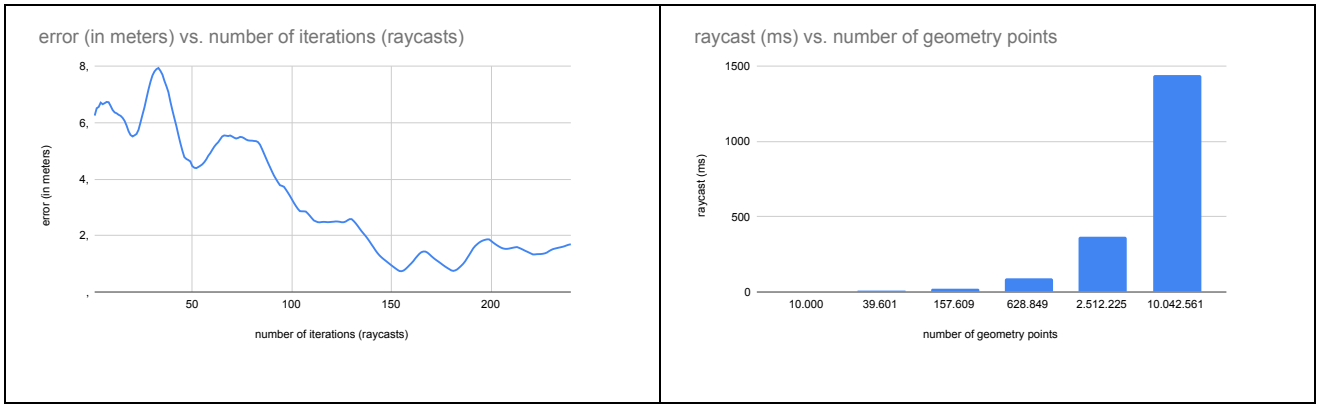

Fig. 4 – Error (in meters) depending on the number of raycast iterations (left). Raycast speed depending on the number of geometry points of the terrain (right).

wind and 0.1-4.3 windless) and roll (0-3.1 with the wind and 0.4-3.3 windless), which in principle follows the normal distribution. The exception is the distribution of the drone's roll in windy conditions, confirming that during calm weather, the error in telemetry is caused by the vibration of the drone itself and in windy conditions by the rotation in the wind direction.

The measurement showed a telemetry error range of a maximum of 5.8 degrees (for pitch, with the wind), so the error in the simulator was modeled using uniform noise with a total variation of 6 units (±3 meters for position, and ±3 degree for rotations).

Simulating the telemetry error (Fig. 7) allows the required number of raycasts to be measured to achieve the required accuracy by averaging. The error at the start is not large (~6 meters) but grows to ~8 m after about 40 iterations (Fig. 8, left). Around the 150th iteration, the error is less than 1 meter, after which it grows again. From this, we can conclude that multiple raycasting solves the problem of unreliable telemetry and that there is an optimal number of required iterations, depending on the telemetry noise of a specific drone (which can be measured and applied in the simulator).

It can be seen from Fig. 8 (right) that the raycast speed can be significantly increased by reducing the number of geometry points (raycast on the terrain with 10M points takes 1441 ms, and with 10K points only 1.99 ms), and this can be done (without significant loss of relief details) by adaptive remeshing and/or using terrain chunks.

The simulator was constructed in SideFX Houdini Apprentice 17.5.391 [18], and simulations were performed on a MacBook Pro (2.7 GHz Intel Core i7, 16 GB RAM, NVIDIA GeForce GT 650M 1 GB).

V. Conclusion

Applying person detection in SAR missions is a significant life-saving aid, but person localization is equally important. The person localization can be realized from the data stored in the drone's camera image. The measurement accuracy of sensors built into the drone can seriously affect the accuracy of localization. In this paper, we have examined approaches to determine a person's location and distance from the drone using the raycast method. We tested the proposed method by using a custom-made procedural simulator. To solve the problem of unreliable telemetry, we calibrated the simulator using telemetry data from images taken by drone, performing multiple raycasts. Depending on the telemetry noise of a specific drone, there is an optimal number of required raycast iterations. In the near future, we plan to use the object detector and parameters of this simulation to study localization performance in real environments.

Acknowledgment

This research was supported by Croatian Science Foundation under the projects IP-2016-06-8345 “Automatic recognition of actions and activities in multimedia content from the sports domain” (RAASS) and IP-2018-01-7619 “A Knowledge-based Approach to Crowd Analysis in Video Surveillance (KACAVIS) and by the University of Rijeka (project number 18-222).

References

[1] S. Sambolek and M. Ivasic-Kos, “Automatic person detection in search and rescue operations using deep CNN detectors,” IEEE Access, vol. 9, pp. 37905–37922, 2021, doi: 10.1109/ACCESS.2021.3063681.

[2] A. Leu, D. Aiteanu, and A. Gräser, “High Speed Stereo Vision Based Automotive Collision Warning System,” 2012.

[3] L. Xiaoming, Q. Tian, C. Wanchun and Y. Xingliang, “Real-time distance measurement using a modified camera,” 2010, doi: 10.1109/SAS.2010.5439423.

[4] J. Sun, B. Li, Y. Jiang, and C. Y. Wen, “A camera-based target detection and positioning UAV system for search and rescue (SAR)

purposes," Sensors (Switzerland), vol. 16, no. 11, 2016, doi: 10.3390/s16111778.

[5] J. Suziedelyte Visockiene, R. Puziene, A. Stanionis, and E. Tumeliene, "Unmanned Aerial Vehicles for Photogrammetry: Analysis of Orthophoto Images over the Territory of Lithuania," Int. J. Aerosp. Eng., vol. 2016, 2016, doi: 10.1155/2016/4141037.

[6] S. Tuohy, D. O'Cualain, E. Jones, and M. Glavin, "Distance determination for an automobile environment using inverse perspective mapping in OpenCV," in IET Conference Publications, 2010, vol. 2010, no. 566 CP, doi: 10.1049/cp.2010.0495.

[7] H. A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," Biol. Cybern., vol. 64, no. 3, 1991, doi: 10.1007/BF00201978.

[8] M. A. Haseeb, J. Guan, D. Ristić, and A. Gräser, "DisNet : A novel method for distance estimation from monocular camera," 10th Planning, Percept. Navig. Intell. Veh., 2018.

[9] J. Zhu and Y. Fang, "Learning object-specific distance from a monocular image," in Proceedings of the IEEE International Conference on Computer Vision, 2019, vol. 2019-October, doi: 10.1109/ICCV.2019.00394.

[10] S. D. Roth, "Ray casting for modeling solids," Comput. Graph. Image Process., vol. 18, no. 2, 1982, doi: 10.1016/0146-664X(82)90169-1.

[11] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016, vol. 9905 LNCS, doi: 10.1007/978-3-319-46448-0_27.

[12] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and physical simulation for autonomous vehicles," arXiv. 2017, doi: 10.1007/978-3-319-67361-5_40.

[13] S. Krishnan, B. Boroujerdian, W. Fu, A. Faust, and V. J. Reddi, "Air learning: An ai research platform for algorithm-hardware benchmarking of autonomous aerial robots," arXiv. 2019.

[14] P. Solovev et al., "Learning State Representations in Complex Systems with Multimodal Data," arXiv. 2018.

[15] M Burić, G Paulin, M Ivašić-Kos, „Object Detection Using Synthesized Data", ICT Innovations 2019, Ohrid

[16] "Phantom 4 Advanced - Specs." https://www.dji.com/hr/phantom-4-adv/info#specs (accessed May 05, 2021).

[17] S. Sambolek and M. Ivasic-Kos, "Person Detection in Drone Imagery," 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech), 2020, pp. 1-6, doi: 10.23919/SpliTech49282.2020.9243737.

[18] "Houdini Help." https://www.sidefx.com/docs/houdini/ (accessed May 05, 2021).