International Journal of Semantic Computing
Vol. 17, No. 3 (2023) 1–24

DOI: 10.1142/S1793351X23620015

World Scientific
www.worldscientific.com

Towards Gesture-Based Cooperation with Cargo Handling Unmanned Aerial Vehicles

Marvin Brenner* and Peter Stütz†

Institute of Flight Systems
University of the Bundeswehr Munich (UniBwM)
Neubiberg, Germany
**marvin.brenner@unibw.de*
†*peter.stuetz@unibw.de*

Received 12 October 2022
Revised 15 April 2023
Accepted 30 April 2023
Published 3 August 2023

This work provides the fundament for a gesture-based interaction system between cargo-handling unmanned aerial vehicles (UAVs) and ground personnel. It enables novice operators to visually communicate commands with higher abstractions through a minimum number of necessary gestures. The interaction concept intends to transfer two goal-directed control techniques to a cargo-handling use case: Selecting objects via deictic pointing communicates intention and a single proxy manipulation gesture controls the UAV's flight. A visual processing pipeline built around an RGB-D sensor is presented and its subordinate components like lightweight object detectors and human pose estimation methods are benchmarked on the UAV-Human dataset. The results provide an overview of suitable methods for 3D gesture-based human drone interaction. A first unoptimized model ensemble runs with 7 Hz on a Jetson Orin AGX Developer Kit.

Keywords: Human-drone-interaction; UAV; pointing; cargo; flight control; gestures; cooperation.

1. Introduction

Unmanned aerial vehicles (UAVs) have shown the potential for high impact in the field of distribution logistics. Their use cases range from covering short gaps between ships in vertical replenishment [1] to long range delivery of medical goods [2] like blood supplies [3]. The global navigation satellite system (GNSS) provides a reliable way of navigating large distances but last mile deliveries [4] and unloading cargo in the wild remains a complex problem to fully automate [5]. A recent UAV navigation literature review [6] found that only 16% of GNSS-independent [7] UAV navigation research contributes solutions attempting fully autonomous navigation rather than working on required subordinate functionalities. A delivery zone in the wild can seem

*Corresponding author.

suitable topology-wise but in reality, be dangerous due to dynamic obstacles or restricted for UAVs to land on. Additionally, heavier payloads might neither allow being dropped onto a predetermined zone [8] nor even be detachable without a human in the loop. Other ambiguous examples include for example a firefighter's driveway, a frozen lake, when the assigned landing zone is occupied or the only safe flight path to the drop zone is blocked. In cases where no safe landing zone can be determined by the UAV itself, guidance by an operator is required. Otherwise, the mission has to be aborted, since draining battery levels while searching could lead to an emergency landing in unknown terrain. Even for fully automated solutions it is likely that due to safety concerns the ability to understand utterances by a human will remain one of the required functionalities. At least until UAV navigation methods are able to develop a comprehensive scene understanding and their motion planning has acceptable error rates across a variety of more complex scenarios in the wild. With rising levels of automation, it can however still be expected that there is going to be a gradual role shift of UAV operators to collaborators. Instead of monitoring the aircrafts constantly, they are increasingly going to be taking supervisory functions for multiple drones and interact with companion-like UAVs [9]. The resulting reduction in mental workload could significantly increase the safety and efficiency of cargo handling. Although visual signaling during complex landing procedures with cargo requires training of specialized personnel [10], it has already become an established technique [11] of aircraft pilots and landing signal personnel. A gesture-based human-drone interaction interface could serve as a similar method for guiding cargo UAVs near ground level and determining a suitable landing zone in the wild. It also has the potential of eliminating the need for additional remote controls and datalinks between operator and aircraft. Remotely operating an UAV requires expert knowledge and a stable communication link with low latency which cannot always be guaranteed for longer distances. Since situational assessments are also more accurate on-site, it is suggested that the UAV uses a gesture-based interface which can be used by non-expert ground personnel at the delivery zone. According to [12], the main issues for integrating aerial cargo deliveries into existing infrastructures are not only technical challenges but also safety and privacy concerns. Enabling UAVs to understand simple hand gestures of non-expert operators and bystanders can help building the public's trust to aerial delivery systems and reduce the required training time to use them.

This work aims to contribute the foundations for a visual processing submodule that can be attached to multi-rotor UAVs and unmanned helicopters. The submodule is intended to be used in the process of navigating a landing zone near ground-level within Very-Low-Level airspace (VLL) below 150 m. All processing will be handled onboard the UAV with an embedded computing device running a marker less, gesture-based interface with access to higher-level functions relevant to a cargo transportation task. It maintains a compact input gesture vocabulary for first-time users through the use of interaction metaphors. Additionally, a visual processing pipeline for the system is suggested along with first fundamental evaluations of

required components like object localization, human pose estimation and gesture classification. It will give human operators the ability to interrupt the UAVs current task, give higher level commands and control its position manually when unloading cargo in dynamic and non-ideal landing zones.

2. Related Works

In the field of human drone-interaction (HDI) gesture-based interfaces to control the attitude of UAVs have been extensively researched in the last decade [13]. The majority of reviewed approaches concentrate on using body [14] or hand gesture inputs [15] as an alternative flight control method to remote control (RC) joysticks. Typically, the UAV's six degrees of freedom are mapped to six gestures in order to enable gesture-based piloting. Whenever a system can perform additional commands, user-centered researchers commonly apply strategies like gesture elicitation [16] where users are asked to suggest gestures for a given function and the final input vocabulary is selected on the base of agreement scores [17]. Alternatively, gestures can be designed by experts based on known visual signals [18] like the NATOPS [11] aircraft signals or sign language. Although intuitive user-defined gestures can decrease mental workload for operators, they do not necessarily improve task performance [19] or result in a coherent vocabulary [20]. This should be evaluated by providing a simulated model for the UAV which can react to user inputs. It can be achieved through Wizard of Oz methodology [21] as a first step, then further developed with virtual environments build with engines like Unity [22]. When it comes to flight control with gestures, the UAV's flight dynamics should be modeled and additionally fed into the simulation to get a preliminary impression on how the UAV will behave when conducting real flight experiments [23]. Conventional gesture interfaces for UAV flight control typically require eight gestures for roll, yaw, pitch and horizontal rotation. Adapting these one-to-one mappings to tasks that require additional commands can lead to the gesture vocabulary exceeding reasonable dimensions and mixing of multiple control metaphors. This expansion not only sacrifices the initial ease-of-use of a gesture interface but also increases cognitive load [24] for operators compromising flight safety. It induces higher risk of false-positive detections and human error compared to RCs that rely on a single control metaphor. Until recently [14], gesture-controlled interfaces commonly did not give the pilot control of the UAVs velocity due to safety concerns. Although this may decrease the risk of accidents, it can render tasks tedious and inflate their duration. There appears to be a gap between current HDI research and practical interfaces for professional drone pilots, in particular for use cases other than aerial photography.

One-to-many gesture mappings using metaphors could be a solution to the problem of growing gesture vocabularies, as their coherent structure improves memorability [20]. Deictic pointing gestures [25] have been used frequently in HDI to direct attention towards a target object [26]. They can be easily performed and understood from a young age [27] while serving as a flexible tool to select areas or

objects of interest [28]. When using an inertial-measurement-unit (IMU) wearable sensor armband, pointing gestures can be used to direct UAVs to a location and select them in a multi-drone scenario [29]. For close distances up to six meters, small drones can even move to a pointed target location with less delay and flight route variance [30] compared to joystick control. In [31], a firefighter can point towards the windows of a burning building which the UAV should enter to inspect the selected room. This is achieved by extracting skeleton points from the image and detecting intersections of the pointing arm vector with detected objects in the image. Since the UAV needs to distinguish vertically aligned windows, a form of depth estimation is required as well. The authors of [32] achieved this by using the sparse point cloud of a Simultaneous Localization and Mapping (SLAM) method. Known target objects can then be selected via a pointing arm vector directly within image space. This ability can unlock new interaction contexts without increasing the amount of necessary gesture inputs. Pointing a cargo UAV to objects and locations of interest can be a useful interaction method but certain transportation tasks additionally require precise landing procedures if no visual markers on the ground are available. Therefore, a second gesture control strategy is required which additionally lets the user pilot the UAV without sacrificing safety compared to remote controls.

Overall, the technical challenges of deploying a gesture-based interface onboard of cargo UAVs with limited power supply, payload weight and computational resources become apparent: (1) The embedded computing device onboard needs to be capable of running object detection, human pose estimation and a form of depth estimation simultaneously with low response times. (2) Input commands via gestures are prone to inaccuracies from all three methods as well as human error. (3) In order to select objects in image space, the UAV needs a wide field of view to cover the scene but sufficiently high resolution in regions of interest (ROI) to detect landmarks.

Comparison to the conference version

This work is an extension of [33] and improves the preliminary publication by the following two additions:

- The benchmark on the UAV-Human Dataset is extended by nine new variants of light-weight object detectors and one vision-transformer method for human body key point estimation which requires a bounding box as a prior. Provided the new results and deeper analysis, common components of the highest performing architectures are pointed out and additional visualization is provided for better overview.
- For direct flight control and pointing, accurate 3D key points are required, therefore we compare the positional and angular error of three body key point lifting methods to the inside-out tracking of two Oculus Rift S controllers which represent the ground truth.

3. Method

3.1. *Concept description*

Assuming the cargo UAV has already navigated to a delivery area via GNSS, it has to search for suitable landing zones to deliver the cargo. As soon as the UAV is descending close to ground level, it needs to be able to monitor its surroundings to maintain safety distances. It can automatically enforce the 1:1 rule for example which states that as a safety requirement, the UAV should keep a lateral distance of at least its own flight height in meters to any person on the ground. Although it is possible to detect and track dynamic objects, the risk of causing damage might be too high for the UAV to land alone. It should then try to detect an operator and constantly monitor battery levels in case there is no guidance for a landing available immediately. Once a person is walking towards the UAV, it should turn the camera towards them and check whether or not they are trying to request authentication. Once an operator was successfully verified, it should be able to follow given hand signals for guidance until it can drop the cargo safely. Afterwards it can be guided towards the next cargo package for pick up. If it is equipped with a pick-and-release mechanism or aerial manipulation tools like robotic arms [34], it can pick up the cargo autonomously. Otherwise, it tries to land close to the cargo and the ground personnel can attach a new payload. Despite monitoring its surroundings, the UAV's sensors should not lose track of the operator when ascending.

3.2. *Visual processing pipeline*

Monitoring the surrounding environment with an onboard multi-camera-system brings additional weight, bandwidth and power requirements onto the already limited UAV platform. A small dual fisheye camera could be an efficient solution to achieve surrounding vision, since there are already commercial off-the-shelf sensors available with automatic in-camera stitching of image borders and perspective transformation into a panoramic view. Due to the 360° field of view (FOV), the number of pixels per degree is significantly lower both vertically and horizontally compared to a sensor of equal resolution that has a smaller FOV. Especially near image borders with higher distortion, low resolution in regions of interest degrades human pose estimation performance [35] at larger distances significantly. The majority of heatmap-based approaches suffer both from low resolution and joint quantization error [36]. Therefore, this submodule builds on a peripheral-vision camera setup inspired by [37] where the omnidirectional camera is combined with a gimbal supported high-resolution camera featuring a tele lens to classify aircraft threats. Since this use case is close to ground level, the latter camera is replaced with a passive high-resolution stereo depth sensor. The chosen device has a focal length of 4 mm and a baseline of 12 cm, an integrated IMU and a polarizing filter to reduce the glare of sunlight. Although stereo depth error grows quadratically with increasing distance, it is hypothesized that the accuracy could be sufficient to distinguish

relative depths between objects within a range of 40 m and maintain safety distances to surrounding bystanders in proximity. For larger distances, a gimbal camera with controllable optical zoom has to be used. Stereo RGB-D was chosen over a LIDAR sensor to keep weight, cost and power usage low. At the same time, it provides semantic information that can be easily matched and synchronized with depth estimates. Feedback-wise, the gimbal supported sensor can also serve as an indication of the UAV's focus of attention. For example, when the UAV is commanded to relocate, it turns the main sensor in the corresponding direction in anticipation before actuating the rotors. It can also turn towards target objects to give feedback on the object selection without losing track of the operator. This way, the operator can recognize undesired behavior early and prevent it with a gesture. Additionally, a LED matrix attached to the UAV can display the current interaction context [23]. Both sensors are connected to an embedded computing device — here a NVIDIA Jetson Orin AGX developer kit is intended to be used — for processing inputs. Since the perception framework needs to communicate with the flight control system of the UAV platform, the robot operating system (ROS) is chosen for intra-process communication. The system receives camera images, IMU and GNSS data as an input and sends a command with flight parameters according to the protocol of the flight control system as an output. Analogue to the camera setup, the visual processing pipeline can be subdivided into central and peripheral vision processing (compare Fig. 1). The peripheral image stream of the omnidirectional camera is used to keep track of the UAV's movement as well as to detect surrounding objects. On the central vision stream the landmarks from objects of interest and their depth estimates are acquired. This is similar to human vision, where peripheral vision appears

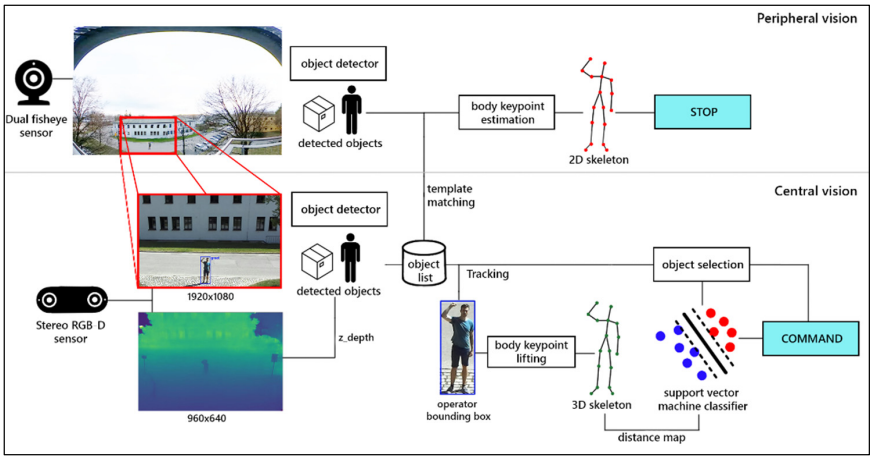

Fig. 1. The proposed onboard system architecture consisting of a peripheral-vision camera setup connected to an onboard embedded system on module (SOM) which takes care of visual processing. The pipeline is split into two branches of peripheral and central vision processing.

to be more useful for the perception of self-motion and central vision tends to be used for analyzing patterns and detecting the motion of objects [38]. If the cargo payloads, landing zone markings and other relevant entities like pedestrians are known *a-priori*, a single-stage object detector can be trained to extract the coordinates of the corresponding bounding boxes from the peripheral image stream. If more accurate segmentation masks are required, the detector could also be extended with a semantic segmentation method that returns fine-grained per-pixel segmentation masks.

Twelve required functions for a transportation use case are defined: taking off, landing, returning to launch, authenticating the current user, transfer flight control, stop any activity, follow the operator, follow a person, move to a location, move in a direction, pick up an object, drop the current object, transport an object to a location and negating a command. Any detected person can perform a greeting gesture by raising their hand above head level in order to request authentication. To classify this gesture, their pose represented by skeleton joints is acquired through human pose estimation. Generally, human pose estimation can be performed either bottom-up by detecting body key points in the image and organizing them into distinct skeletons or top-down by first detecting the bounding box of a person and then estimating the position in pixels of all key points inside that region. Once they are available, waving at the UAV on ground level can already be detected by a simple fuzzy logic check of the hand joint coordinates. The greeting gesture triggers the rotation of the central vision camera (CVC) towards the operator. The new gimbal angle is based on the coordinates in the image space of the omnidirectional view. For verification of identity, the UAV can scan a fiducial marker on a card or use facial recognition with a database of known operators. After successful authentication, each person is tracked with SORT [39] and the operator is additionally distinguished by color histogram matching. For any detected object in the CVC's image view, depth value samples from the point cloud can be accessed to estimate its distance to the camera. The two-dimensional body key points of the operator's skeleton are already sufficient to classify gestures but not to enable direct flight control via proxy manipulation. Determining which object an operator is trying to select when multiple potential candidates intersect with the pointing line, also poses a problem.

Acquiring the z-depth of skeleton joints can now be achieved by one of the following three lifting methods: First, using the pixel locations of the joints in the image, the depth values from the RGB-D sensor can be accessed. Second, the 2D key points can be compared to a library of 3D poses via proximity search. Thirdly, a deep neural network can learn to regress the 3D key points from a 2D key point input with an accurately labeled dataset. Each method heavily relies on accurate 2D key points as a prior and needs to be compared to a ground truth in order achieve an optimal trade-off between precision and latency. Which method is preferable will depend on the background noise as well as the distance between camera and operator. The acquired body key points can be encoded with a distance map and used as a feature vector for a support-vector machine classifier.

3.3. *Gesture vocabulary*

The proposed gesture vocabulary (Fig. 2) is chosen with a transportation task near ground-level in mind where an operator can either directly control the UAVs flight or give highly abstracted commands [40]. The combination of all possible commands for a transportation task into one metaphor is not feasible, therefore two flexible and intuitive metaphors that can be mapped to multiple commands were chosen. The two core concepts for interacting with the UAV are deictic pointing gestures to select task-relevant objects and a proxy manipulation gesture for direct flight control (DFC). Apart from answering permission requests and flight control, all highly abstracted commands are executed via a pointing gesture. At any moment the operator and bystanders can raise both hands above head-level to interrupt the current command in case of undesired behaviors. Allowance to perform a maneuver is given by not performing a stop gesture within and remaining neutral for a predetermined duration once the UAV has displayed the requested command. Once it is verified that the system is sufficiently accurate and an operator is more experienced, this duration can be reduced since stopping the UAV is possible at any given point in time. Apart from stopping the UAV, all commands can only be given by a person who was already authenticated as an operator. Requesting authorization via a gesture is only possible while there is no other operator detected. The overall design is determined by the interaction contexts, ergonomics and the system's gesture recognition performance. Even though this is the first expert-led iteration of the gesture vocabulary, cognitive attributes like discoverability, intuitiveness and learnability for the end user need to be kept in mind [41]. Therefore, a minimum number of gestures in a coherent structure are used, where each of the three contexts has its own mental model of gesture-based interaction. This way, each gesture can allow more variations without creating overlaps with other commands.

3.3.1. *Object selection via deictic pointing*

As soon as a pointing gesture is detected, the interaction context is switched to "Selection", where no gestures apart from pointing and the "STOP" gesture are recognized. Detected cargo goods, landing zones, humans and the UAV itself can now be selected by the operator via a pointing gesture. The selection is based on a pointing ray originating from the operator's head and passing through the pointing hand. Alternatively, it can originate in the shoulder or be a mix of both vectors [26] but a more sophisticated solution requires further experimentation for aerial scenarios. The target is then selected based on a ray cast hit return value, its distance to the operator and whether it fits the interaction context. It needs to be determined whether absolute or relative ray casting via the arms is a sufficiently accurate selection method for aerial perspectives and how it can be improved through heuristics. Pointing can trigger five possible behaviors, the first one is simply selecting an object by pointing for a time t at the object. The object is added to a set representing the

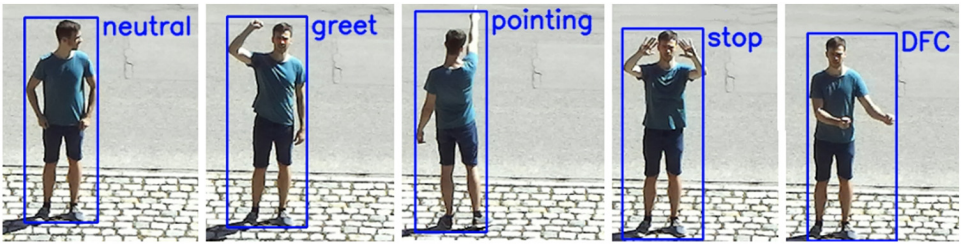

Fig. 2. The initial simplistic gesture vocabulary with five main gestures.

selection. If there is only one object in the selection, the UAV will simply give information about direction and color of the selected object to ensure the operator selected the right one. As soon as two or more unique objects are in the set, the UAV can suggest a command to the operator based on the selection. For example, if the UAV itself and a cargo payload are in the set, the UAV will ask the operator if it should pick up the selected object. Abstract commands like "Transport this object to that location" can be given with a selection of {UAV, OBJECT, LOCATION}. Because the action space in this use case is rather limited, each action can be mapped to a unique selection set. Performing a stop gesture, clears the selection set and reverts the interaction context back to "Idle". Taking off is activated by selecting the UAV while it stands on the ground and then confirming the take off.

3.3.2. *Direct flight control utilizing proxy manipulation*

The DFC context is initialized if the operator holds both hands in front of them at the same height with a gap of about one elbow length. Now the operator can control the UAV's attitude by turning, pushing and pulling an imaginary proxy between both hands. The proxy manipulation relies on receiving viable 3D joint positions of both hands. If the DFC gesture is held for three seconds, the midpoint between the hands is used as an initial reference point. As long as the gesture is active, any deviation of the hand midpoint from the reference point along the x, y and z-axis will result in the UAV moving in the same direction, relative to the operator's viewpoint (Figs. 2(a) and 2(b)). The magnitude of the vector from the initial reference point to the current hand midpoint is then deemed proportional to the UAVs acceleration in the corresponding direction, limited by a maximum threshold. An initial line from the right to the left hand which is orthogonal to the facing direction of the operator is used to manipulate the yaw angle rotation of the UAV. The angular deviation from the initial line when moving the hands then determines the yaw angle gain (Fig. 2(c)). With proxy manipulation, the translation, yaw attitude and velocity can be controlled using only a single gesture. If the operator tries to move the UAV downwards while its height falls under a predefined threshold, it suggests landing at that position. The lateral movement changes to the side can be easily observed by the operator but movement speed forwards and backwards as well as up and down can be

indicated by a 2D animation on the LED display. To get a first impression on the basic functionality of the interaction model, it was implemented in a virtual environment (VE) using Unreal Engine and a head-mounted display (HMD) with two hand controllers. A simple ray cast function to select an object was implemented as well as the proxy manipulation of a 3D multi-rotor UAV model based on the HMD and hand controller transforms in the scene. Switching between direct flight control and pointing gestures worked seamlessly in the VE.

4. First Experimental Results

Outside virtual environments the precision of controlling the UAV's flight with 3D gestures is affected by the CVC's depth error which needs to be evaluated for long distances of up to 40 m. It needs to be ensured that via ray casting from the arm joints, objects can be selected by the operator within the CVC's point cloud. To detect objects in the panoramic image stream, a fast detection method with high recall for detecting humans in particular is required. A light-weight human pose estimation method with reasonable joint localization error needs to be determined, ideally with the ability to lift the resulting body key points to 3D. Finally, the overall system architecture for the complete process of gesture recognition must be precise and lightweight enough to be deployed on the embedded system. In previous work [33] the mean absolute error of the CVC depth map for a resolution of 2208×1242 was determined by placing a checkerboard of size 60×80 cm and with a square width of 10 cm at varying positions at a distance between 1.5 and 40 meters. The distance estimation was then compared to measurements of a 650 nm laser range finder with an error of 1.5 mm. Outdoors the mean absolute depth error for a distance of 1.5 to 35 m was 1.2 m with standard deviation of 1.25 m. The overall results were in line with the manufacturer's claim of less than 2.1 m of depth error below 30 m. It can be expected that up to 15 m, depth error remains under 50 cm. For object selection via ray casting in this range, the precision can be accepted as long as objects are not placed closely together. For larger distances, the depth values can only roughly be used to order objects relative to each other and depth estimation has to rely on monocular cues like textures or known object dimensions.

4.1. *Body key point localization on the UAV-human dataset*

While the number of datasets with aerial imagery and annotations for object detection has increased recently, the process of acquiring ground truth labels for human pose estimation is time-consuming, especially for three dimensional joints. But even for body key points in image space, there is a lack of datasets with annotations that were recorded with elevated viewing angles while flying. Current datasets often only feature a small number of unique poses determined by a predefined number of gestures [42] or lack variety regarding location, distance, height and lighting conditions [43]. Therefore, it was decided to use the challenging UAV-Human [44] dataset for a

preliminary benchmark of available off-the-shelf object detectors and human pose estimation approaches. The dataset contains labels for action recognition, human pose estimation, attribute recognition and person re-identification. It was recorded with a UAV flying at varying speeds and heights from two to ten meters and contains 22476 frames that were labelled for human pose estimation by 15 volunteers. Its backgrounds are rich in variation featuring indoor and outdoor scenery in urban and rural areas which is either lit by various intensities of sunlight or artificial lighting. For this benchmark, all evaluated approaches were selected based on low inference time, their reported average precision on the COCO [45] 2017 validation test-set as well as on their ability to deal with low resolution images. The used COCO AP metric considers objects small when their area in pixels is 32*32 or smaller, medium if their area is between 32^2 and 96^2 and large if their area is larger than 96^2 pixels. Since UAV-Human has a resolution of 1920×1080 pixels instead of 640×480, the object size category thresholds used here were scaled accordingly. After scaling, 4.2% of samples contain small patches of humans, 80.13% medium and 15.67% large sized ones. Inference times were computed for passing forward a single batch on a NVIDIA RTX Mobile A3000 GPU averaged over 20.000 iterations. The time needed for pre-processing, loading the model and visualization is not included. Since the top-down human pose estimation approaches require bounding boxes and one method is not applicable for multi-person samples, the first benchmark is for testing object detectors in their ability to detect humans on the UAV-Human Benchmark without any

Table 1. Human detection on the UAV-Human dataset.

Architecture	Ref.	Params (M)	Input size (px)	mAP (%)	mAP50 (%)	mAR (%)	Mean IOU	Latency (ms)
EfficientDet Lite 0	[46]	3.2	320	42.2	78.1	19.9	0.60	12.9
YOLOX-Nano	[47]	0.9	416	47.1	75.7	28.9	0.62	**7.1**
Efficient-Det 0	[46]	3.9	512	52.8	75.3	38.0	0.61	15.4
NanoDet+-m-1.5x	[48]	2.4	320	57.8	88.9	46.8	0.71	8.4
YOLOX-Tiny	[47]	5.1	416	59.0	88.5	41.7	0.70	7.3
YOLOX-S	[47]	9.0	640	59.9	89.5	42.3	0.71	7.8
NanoDet+-m	[48]	1.2	416	61.6	93.9	47.7	0.74	8.3
NanoDet+-m-1.5x	[48]	2.4	416	62.4	**94.7**	49.1	0.75	8.6
RTMDET	[49]	4.8	640	63.2	83.8	38.7	0.69	14.5
YOLOR-CSP	[50]	52.9	640	63.9	83.1	38.2	0.71	49.6
KAPAO-S	[36]	12.6	1280	64.0	88.7	43.4	0.72	8.6
YOLOv7	[51]	36.9	640	65.7	85.3	41.5	0.71	22.4
YOLOR-CSP-X*	[50]	52.9	640	66.1	84.9	40.9	0.71	32.0
DAMO-YOLOT*	[52]	8.5	640	69.0	90.4	48.9	0.75	10.7
YOLOR P6	[50]	37	1280	69.1	89.4	47.5	0.75	57.7
DAMO-YOLOS*	[52]	16.3	640	**69.9**	91.2	**50.4**	**0.76**	13.0

Notes: The columns from left to right refer to the method's variant name, the authors, the number of parameters in millions, the input image width in pixels, the mean average precision across all 10 IOU thresholds, the mean average precision for IOU ø= 0.5, the mean average recall across all 10 IOU thresholds, the mean intersection over union values compared to the ground truth bounding boxes and finally the latency for the used setup in milliseconds.

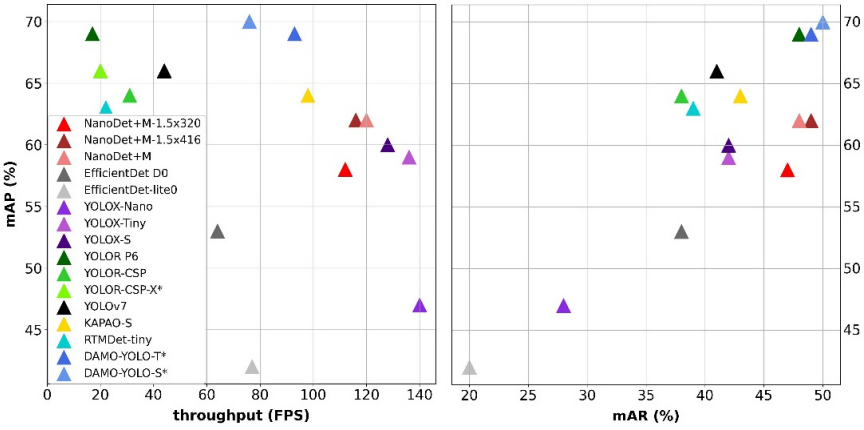

Fig. 3. Human detection on the UAV-Human Dataset. The y-axis shows the mean average precision across the 10 COCO intersection over union thresholds. On the left, the precision is compared to the throughput in processed frames per second on a RTX A3000 Mobile GPU while on the right the x-axis displays the mean average recall. For the embedded plattform NVIDIA Jetson Orin AGX usually about half the throughput can be expected based on two implemented comparisons so far.

prior training on the dataset itself. Mainly computationally lightweight object detectors are compared regarding precision and recall because all the consecutive processing steps depend on accurate localization of humans in image space. Simultaneously a low latency is required as the detector has to predict bounding boxes over the full omnidirectional view in a reasonable resolution.

Human detection is evaluated via the intersection over union metric (IOU) where the area of intersection between the ground truth and the predicted bounding box is divided by their area of union. The provided skeleton joint position labels of the dataset were used to create bounding boxes as a ground truth for human detection. Using the minimum and maximum joint positions in x- and y-direction results frequently in a tight box omitting the arms or feet, therefore all bounding boxes were increased by 20% in width and 15% in height to wrap tightly around the whole person. The COCO average precision and recall metric is the mean over ten IOU thresholds in the interval $\{0.5, 0.55, \ldots, 0.9, 0.95\}$. Every object detector was allowed a maximum of 10 detections while the lowest accepted confidence was set to 50%. The results indicate that the new addition of DAMO-YOLO-S [52] is currently the most efficient trade-off for human detection on UAV-Human. Not only did it reach the highest precision and recall over the large-scale YOLOR P6 [50] but it also has a competitive latency of only 13 miliseconds (ms) on the used setup. YOLOX-Nano [47] has the lowest latency with 7.2 ms but its precision is second to worst due to a lack of complexity in order to learn more generalizable representations. While the nanodet+m-1.5x variant outputs less precise bounding boxes, it still has a solid recall score of 49% at a processing speed of only 8.6 ms. The RTMDet [49] architecture achieved state of the art precision on two major datasets for aerial object detection,

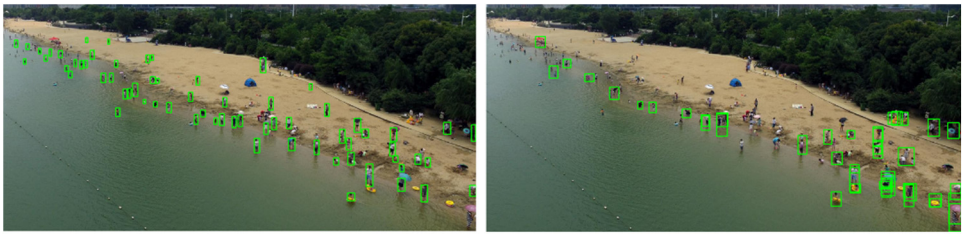

Fig. 4. Qualitative side-by-side comparison of KAPAO-S (left) and NanoDet-m-1.5x detection bounding boxes on the TinyPeople Dataset. KAPAO-S achieves precise localization despite low confidence.

but it has high latency in relation to methods with comparable precision for this dataset. Generally, the top-ranking architectures in this benchmark (Table. 1, visualized in Fig. 3) all incorporate their own feature pyramid technique where the learned representations are fused efficiently across multiple scales to achieve better results on low resolution images without sacrificing inference speed. The small KAPAO [36] architecture variant achieves a mAP of 64% and has the advantage that multi-person body key point estimation is already included in the inference time. It stacks multiple object detectors and formulates key points as objects inside the first detected bounding box which was used for this first evaluation. From a few qualitative comparisons, KAPAO-S even generalizes well to datasets like TinyPeople which was recorded from far higher altitudes, without additional training. When lowering the accepted confidence threshold to 5% for detections, its low confidence predictions are still precise in localization (see Fig. 4) compared to other architectures like Nanodet-m-1.5x. The next human pose estimation benchmark was conducted to determine how precise legs, torso and arms of a person can be localized in image space. For two methods that require bounding boxes as a prior, the detection results of KAPAO-S and DAMO-YOLO were used as an initial region of interest.

Human pose estimation is evaluated for 2D key points, using the COCO [45] mean average precision (AP) metric with Object-Keypoint-Similarity (OKS). A singular key-point similarity $ks(\hat{\theta}_i^{(p)}, \theta_i^{(p)})$ is computed by passing the Euclidean distance between the ground truth key point position $\theta_i^{(p)}$ and the predicted key point position $\hat{\theta}_i^{(p)}$ in image space into an unnormalized gaussian with standard deviation sk_i. The object scale s is the square root of the object area in pixels and the constant $k_i = 2\sigma$ is the falloff determined separately for each of the 17 key point types (nose, arm, shoulder, knee, etc.). The reasoning behind this being that an absolute error of 5 pixels for a knee key point should decrease ks less than for a nose key point because the surface area of the knee is larger.

$$\boldsymbol{ks}(\hat{\boldsymbol{\theta}}_{\boldsymbol{i}}^{(\boldsymbol{p})}, \boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{p})}) = \boldsymbol{e}^{-\frac{\|\hat{\boldsymbol{\theta}}_{\boldsymbol{i}}^{(\boldsymbol{p})} - \boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{p})}\|_2}{2s^2 k_i^2}}, \tag{1}$$

$$\boldsymbol{OKS}(\hat{\boldsymbol{\theta}}^{(\boldsymbol{p})}, \boldsymbol{\theta}^{(\boldsymbol{p})}) = \frac{\sum_{\boldsymbol{i}} \boldsymbol{ks}(\hat{\boldsymbol{\theta}}_{\boldsymbol{i}}^{(\boldsymbol{p})}, \boldsymbol{\theta}_{\boldsymbol{i}}^{(\boldsymbol{p})}) \boldsymbol{\delta}(\boldsymbol{v_i} > \boldsymbol{0})}{\sum_{\boldsymbol{i}} \boldsymbol{\delta}(\boldsymbol{v_i} > \boldsymbol{0})}. \tag{2}$$

Table 2. Human pose estimation on the UAV-Human dataset.

Architecture	Ref.	Params (M)	Input size (px)	mAP (%)	mAP50 (%)	meanOKS	Latency (ms)
KAPAO-S+ BlazePose	[53]	25.3	256	26.9	46.2	0.45	21.7
OpenPifPaf (MNetV3-L)	[54]	5.4	224	40.8	48.8	0.50	23.2
KAPAO-S	[36]	12.6	1280	50.4	61.0	0.54	**8.56**
KAPAO-M	[36]	35.7	1280	51.7	61.5	0.58	19.5
Higher-HRnet	[55]	28.6	512	56.5	—	—	89.1
AlphaPose	[56]	49.2	512	56.9	—	—	59.4
YOLOv7-Pose	[51]	36.9	640	57.3	69.5	0.64	22.4
DAMO+ViTPose-B	[57]	96.5	256	**68.2**	**82.1**	**0.75**	20.5

Notes: Table 2 lists the same columns as Table 1, but the mAP is now based on OKS. The +-operator indicates that the first architecture supplied the prior bounding box while the second one was used to detect the body key point positions within that region.

The key point similarities are averaged only over labeled key points that are visible ($v_i > 0$) to calculate the overall OKS for one person. Again, to determine the COCO average precision and recall, the results over 10 OKS thresholds in the interval $\{0.5, 0.55, \ldots, 0.9, 0.95\}$ are averaged. The visibility index $\boldsymbol{\delta(v_i > 0)}$ is not included in the dataset.

The authors of UAV-Human provided human pose estimation results for AlphaPose and Higher-HRnet but without the details of AP50, AP75 and mean OKS. For comparison, one bottom-up heatmap regression method named OpenPifPaf (MobileNet-V3-Large backbone) and two detector-based methods namely KAPAO-S and YOLOv7-Pose are evaluated. BlazePose is a standalone single-person method which is trained with key point heatmaps but during inference time only uses a regression branch which is supervised by the learned heatmaps during training. UAV-Human occasionally features labels for two humans that both stand in the foreground, therefore the images for BlazePose were split in half at a centered line between the two human's hip joints. Afterwards the method was run separately on each image half but due to relying on an initial face detection, it only achieved 23.1% mAP. One reason for this result is that the faces in the UAV-Human dataset are not always visible or recognizable in lower light situations. Secondly, the method was trained for real-time inference on mobile devices and intended only to be used from a small distance with the person occupying a large fraction of the frame. Despite this fact, the standalone method has a latency of only 7.3 ms and can return segmentation masks and 3D key points through GHUM model fitting. As a consequence, it was attempted to use it on bounding boxes of KAPAO-S which increased its mAP to 26.9% but without replacing the face detector at the start of the model architecture, it is unable to further improve its result. In order to compare the light-weight architectures to the current state-of-the-art in human pose estimation on the COCO dataset, one vision-transformer based architecture called ViTPose [57] was added to the benchmark (Fig. 5). When predicting key points inside the detection results of

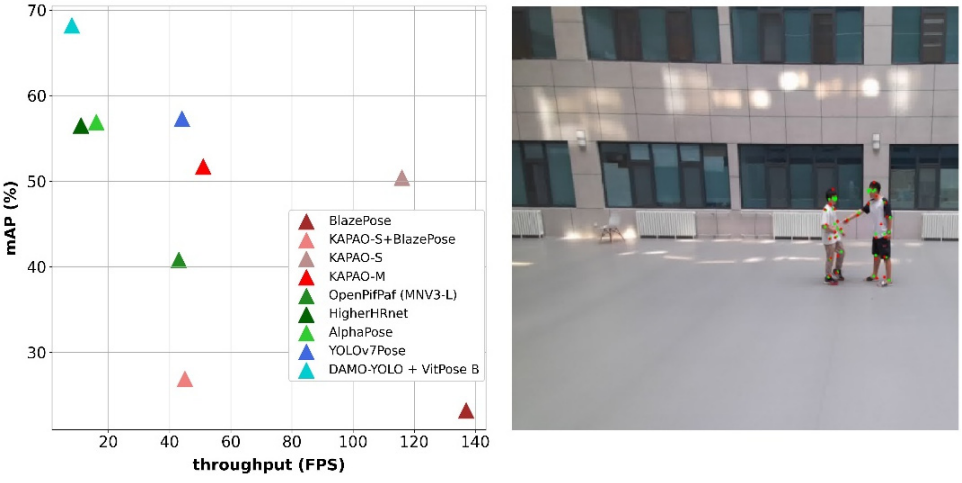

Fig. 5. (Color online) Human pose estimation on the UAV-Human Dataset. Left: A comparison of methods, where the x-axis displays the throughput in processed frames per second on a RTX A3000 Mobile GPU. The y-axis shows the mean average precision across the 10 OKS thresholds. Right: Example image from the UAV-Human with groundtruth joint positions in green and predicted joint positions by OpenPifPaf in red.

DAMO-YOLO, the ViTPose Transformer also achieved the highest mAP by a margin of 11.2 percentage points on UAV-Human. But with the major drawback of the ViTPose-B transformer being its latency of 113 ms, it mainly serves as a comparison. Its new lighter variant ViTPose+-S reaches 84 ms but was not evaluated as it is also not going to be feasible with all the remaining computation on the target embedded system yet. This can be expected to change in the near future, since ViTPose is only a first baseline and vision transformers for body key point detections are still at an early stage of research. Aside from the vision transformer, YOLOv7-Pose reached the highest average precision of 57% but KAPAO-S can still be considered the best trade-off between accuracy and latency for embedded systems.

4.2. *Body key point lifting to 3D*

The two core components of the proposed interaction method also demand depth information for the joints. 3D key points are not only useful to classify gestures more accurately from extreme viewing angles but also necessary in order to realize the proxy manipulation gesture and to be able to point at objects in the point cloud of the stereo depth sensor. Since motion capture systems like Vicon cannot be easily setup in multiple locations, the inside-out positional tracking of an Oculus Rift S is used as a ground truth. The head mounted display (HMD) utilizes multiple wide-angle cameras attached to the headset and controllers with an infrared ring and integrated IMUs [58] to track their position in relation to the headset. In comparison to a Vicon motion capture system, the controller movement tracking can achieve a

translational accuracy of 4.36 ± 2.91 mm and a rotational accuracy of $1.13 \pm 1.23°$ [59]. The HMD can easily be setup at varying distances and works outdoors without occluding the body to a degree where the key points can no longer reliably be estimated in image space. Although the global world frame position of the two controllers' changes for each start of the headset, it was decided to track their relative change in position compared to the head. The Oculus Rift S is calibrated once with a square area of $2 \times 2\,\mathrm{m}^2$ and an HMD height of 1.85 m, it returns the HMD and controller positions in centimeters. The coordinate frame of the headset has the x-axis point towards the front of the headset, the z-axis notes height differences and the y-axis points to the right of the headset to track lateral changes. Since the position estimation of the controllers is determined by accelerometers and tracking of visual markers it is important not to remain fully still in between position changes as the position of the controllers will start to drift away after 2–3 s until they are moved again. For this first experimental setup indoors, one workstation with Unreal Engine was connected to the Oculus Rift S in order to record the timestamped tracking results. Then a laptop connected to the CVC on a two meters high tripod is setup at a distance of five, ten and fifteen meters to observe changes in error across these distances.

KAPAO-S is used to extract 2D key point positions of the head and both hands which are utilized as an input for two lifting methods to get the z-depth: For one, the coordinates are used to acquire the corresponding points from the point cloud in the CVC's coordinate frame which is set to a scaling in meters. For the other one, all key point coordinates are regressed by a two-stage feed-forward neural network [60] trained on the Human 3.6 dataset [61]. The dataset contains 3.6 million different 3D poses of 17 different scenarios like talking on a phone, discussing and pointing. The

Table 3. Body key point lifting from 2D to 3D.

Lifting method	Dist. (m)	X	Y	Z	mDC	yaw	pitch
BlazePose-GHUM	5	0.264	0.507	0.297	0.193	8.32	5.94
[62]	10	0.237	0.535	0.298	0.223	12.65	10.98
Latency: 7.3 ms	15	0.233	0.376	0.237	0.267	13.15	9.75
	Mean	0.245	0.473	**0.277**	0.227	11.37	8.89
Simple-Baseline	5	0.242	0.472	0.308	0.177	8.76	8.33
[60]	10	0.197	0.407	0.282	0.257	6.31	10.05
Latency: 2.3 ms	15	0.172	0.341	0.245	0.213	6.82	6.83
	Mean	0.204	0.407	0.278	0.216	**7.29**	**8.40**
Stereo Depth*	5	0.155	0.447	0.302	0.242	7.03	6.29
Latency: 3.2 ms	10	0.175	0.414	0.834	0.302	7.41	11.50
	15	0.171	0.340	0.793	0.197	8.03	10.98
	Mean	**0.167**	**0.400**	0.643	**0.247**	7.53	9.59

Notes: The reported errors are the average root mean square errors of both recorded hands in meters for x-, y- and z-direction compared to the Oculus Rift S controllers. The yaw and pitch angle errors are reported in degrees. *For the x- and y-directions of the Stereo Depth method, the normalized pixel coordinates of KAPAO-S were used as a comparison, therefore this error has no unit but was still provided for comparison.

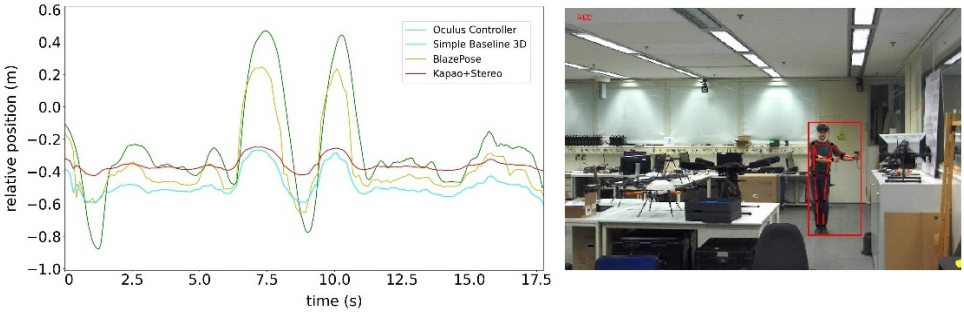

Fig. 6. Relative position change of the left hand on the y-axis in meters for a distance of 5 m.

method returns root-relative joint positions in millimeters and reached a testing error of 62.9 mm when lifting 2D key points that were supplied beforehand by a stacked hourglass network. As a third option — although its key point localization for large distances was not sufficient — it was decided to additionally evaluate how precise the GHUM model-fitting [62] of BlazePose is when the face detector does return a detection based on the bounding boxes by KAPAO-S. It also returns root-relative key point coordinates in meters where the hip joint sits at the origin. The positions of the HMD and both controllers acquired by all methods are transformed by subtracting the HMD position, so that the head sits at the origin and the axis outputs of each method is aligned to the coordinate system of the Rift. Firstly, the absolute angular error in degrees of horizontal and vertical rotations (yaw and pitch) of the arms are determined by calculating the differences in degrees between each estimation method and the oculus controller ground truth. Second, the root mean square error for the hand key point position compared to the ground truth is determined for a distance of 5, 10 and 15 m between camera and person. For each of the three recordings, an identical sequence of movements is performed where the arms are first rotated up and down, then left and right. Afterwards the DFC gesture is used and both hands are moved along x-, y- and z-axis in sequence. Although the procedure is repeated three times in about the same speed for 30 s, the movements are not identical and cause variation for each repetition influencing the overall error. Because the laptop only reached 15 Hz running all three methods simultaneously, 1.350 samples were evaluated for each method, 450 for each distance.

Due to the noise of the stereo depth and occasional outliers in the estimates of BlazePose, these two methods are median filtered with a filter length of five. Since absolute distances on average do not indicate well how the relative movements compare to the ground truth and each method contributes nonlinear noise to the positions, the mean distance correlation (mDC) according to [63] is also reported for all three axes. Before computing mDC, each vector of positions is standardized by subtracting the mean and dividing by the standard deviation. Each distance correlation is calculated by subdividing the distance covariance of both vectors by the product of their respective distance standard deviations. The result is a non-negative

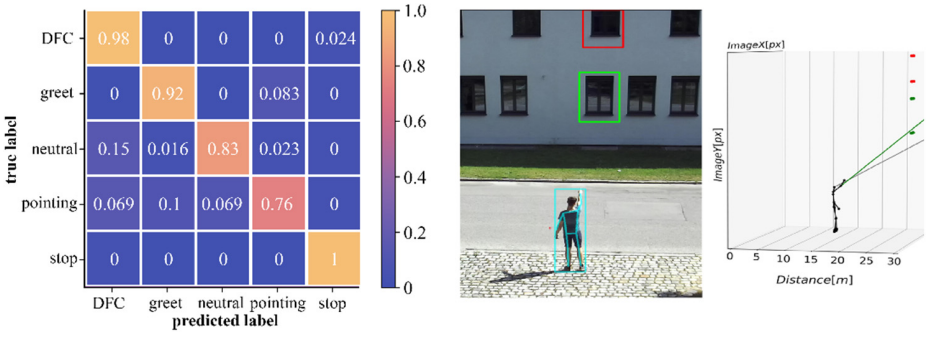

Fig. 7. Left: Confusion matrix of the validation sequence using a SVC with $C = 10$ and radial basis function kernel. Right: The camera was placed 16 m away from the observed person and at a height of 7 m.

value between zero and one, whereas a result of zero indicates that both vectors are independent, and a one signals that they are either identical or variations of one another. Although only a distance up to 15 m was evaluated, all three methods already have a significant positional error of 17–83 cm. As can be somewhat expected for this range, relying on 2D key points by KAPAO and extending them with depth values by the stereo sensor achieved the lowest average positional error for x- and y-direction, even though the distance correlation of 0.247 still has to be improved by additional filtering. For the pointing gestures, the angular error is most important which surprisingly on average was the lowest for the simple baseline approach at 7.8 degrees. In y-direction the error is slightly higher compared to the other two axis which is expected since each method detects the head joint instead of the HMD position in the image. For the z-depth, BlazePose had the lowest positional error on average with 27.7 cm, closely followed by the Simple Baseline method with 27.8 cm.

4.3. *Gesture recognition from elevated viewpoints with a support-vector machine classifier*

For the gesture classification, a small outdoor dataset is collected, where the CVC was placed on four different levels of the 21 m high fire ladder of a building. It records a participant performing five different gestures in random order and from different angles while walking away from the camera. This procedure is repeated four times and the height of the camera is changed from 1.5 to 3, to 6 and then to 12 m. The distance between the observed person and the camera varied between 10 and 27 m. The dataset consists of 3500 frames and is split 1:9 into test and training data, numerical labels are used for the five gesture classes “DFC”, “greet”, “neutral”, “pointing” and “stop”. Because dynamic gestures performed over multiple seconds are slower, have more overlaps and higher complexity due to the temporal dimension, in this approach only static gestures are recognized. Since the sensor will be gimbal stabilized later, depth error and motion blur caused by vibration of the UAV is not considered for this evaluation yet. For each image, KAPAO-S is used to detect the

participant's location in the image and based on the bounding box, BlazePose is utilized to extract 2D and 3D body key points. From the joint data, two feature vectors consisting of all Euclidean distances between the joints are created, one in 2D and one in 3D. All joints are transformed into a coordinate system where the root joint is the origin and are then scaled by the torso height in pixels. Since the majority of poses are linearly separable, a one-versus-all support-vector machine classifier (SVC) with radial basis function kernel is used as a classifier with balanced class weights. K10-Fold cross validation was performed repeatedly with a regularization parameter of C in the range of $\{0.001, 0.01, 0.1, 1, 2, 5, 10, 50, 100, 500\}$. Finally, a C of 10 was used for its highest mean weighted F1-score of 93% for the model-fitted 3D joints and 90% for the 2D joint feature vector. Additionally, a one-minute-long validation video was recorded from 7 m of height and with a distance to the person of 16 m.

All gestures are here performed in a quicker succession like in a real interaction scenario. The UAV was first greeted, pointed towards different locations, and then controlled with the DFC gesture. At multiple occasions, the stop gesture was used in between to cover the transitions from each gesture to the stopping gesture. While the 3D feature vector still achieved a precision of 89%, a recall of 85% and a F1 score of 86%, the 2D feature vector only reached 49% precision, 64% recall and a F1 score of 54%. The confusion matrix of the gestures classified with the 3D feature vector is shown in Fig. 7.

5. Discussion

Using an RGB-D sensor as the main sensor appears to be a lean and versatile option for human drone interaction although it remains uncertain if the depth error of a purely vision-based approach is sufficient to realize the proposed gesture metaphors for object selection and direct flight control. First evaluations of the necessary subcomponents for the proposed visual processing pipeline have resulted in multiple possible options for how to approach the realization of the interaction method. Since the operator needs to be able to point at cargo objects like containers, those need to be detected on both the peripheral and central vision image stream. Nanodet-m is a fast and lightweight detector that can be retrained for this purpose because although it had slightly lower accuracy than KAPAO-S on UAV-Human, it can be expected that its precision will improve once it is trained only for one cargo object class and finetuned on a dataset that features greater heights. In terms of detecting and extracting key points of humans in image space, detector-based top-down human pose estimation methods like KAPAO-S seem to be the best balance regarding localization precision, recall and latency. Although the Nanodet variants have slightly higher recall in this evaluation, KAPAO-S is more robust for detecting humans at larger distances and can precisely localize them in the image even when the confidence of the prediction is lower than 10%. Simultaneously, it can detect 2D key points eliminating the need for additional body key point estimation methods.

Lifting the skeleton joints of the operator up to 3D remains a challenging problem for real-time applications like gesture-based UAV flight control. 2D-to-3D-keypoint-

lifting techniques heavily rely on accurate priors which are still difficult to acquire with low latency on embedded systems. Due the quadratically increasing depth error of the stereo point cloud at distances above 10 m, the model-based lifting approaches seem to deviate less from the ground truth along the z-axis despite less accurate key point priors. One possible solution could be to switch the z-depth acquisition from RGB-D values to a model-based approach once the stereo depth error of the CVC exceeds a threshold that is not acceptable any longer. Additionally, the model-based body key points have to be reliably filtered, since both BlazePose and the Simple Baseline method have high noise and tend to overcompensate slow movements of the hands. While it is possible to improve the object selection process via pointing gestures through heuristics, the proxy manipulation that directly controls the UAV's flight does not allow high positional error. If the tracking in z-direction is not stable after filtering, an alternative gesture which does not compromise the overall gesture metaphor simplicity might be necessary.

The gesture classification performance of the support-vector machine classifier was deemed acceptable for now at 90% because generating additional training data with more variance in viewing angles can still improve the robustness. Any wrong action can be stopped immediately, and safety critical commands still need further additional confirmation. Furthermore, it can be expected that the false-positive-rate decreases when adding a threshold of multiple frames that have to be classified as the same gesture in sequence to trigger the corresponding function.

A preliminary test model ensemble where KAPAO-S and NanoDet-m detect humans and objects every frame while BlazePose extracts 3D joints of one bounding box was implemented on a NVIDIA Jetson Orin AGX Developer Kit. For RGB-D images from the CVC with 1920×1080 resolution, the model ensemble could run with 7 Hz and took around 3.972 of GPU memory. This indicates that the runtime latencies still need to be further improved, for example by reducing the frequency of cargo object detections through tracking.

6. Outlook

An interaction method for gesture-based interaction with cargo-handling UAVs has been presented along with a visual processing pipeline. It relies on an omnidirectional dual-fisheye camera for peripheral vision of its surroundings and uses a gimbal-supported high-resolution stereo RGB-D sensor as a main sensor. The proposed gesture vocabulary is compact and relies on two core metaphors which are deictic pointing and proxy manipulation. With pointing gestures, an operator can select task-relevant objects to give highly abstracted commands like following a person or transporting an object from point A to B. If necessary, the UAV's position can be adjusted via a proxy manipulation gesture which is useful for landing zones without infrastructure and difficult terrain. Three methods for 3D human pose estimation and object detection have been selected based on evaluations on the UAV-Human Dataset. The overall system's main two limitations needing further research are: (1)

the precision of selecting objects with a 3D pointing gesture ray cast technique at longer distances, (2) a control function for mapping proxy manipulation hand movements to the UAV's acceleration.

Acknowledgments

The presented research was supported by the German federal aviation research program (LuFo VI-2, ID: 20D2111D). It was also supported by dtec.bw — the center for digitalization and technological research of the Bundeswehr. dtec.bw is financed by the European Union — NextGenerationEU. The APC was provided by the Universität der Bundeswehr München (UniBwM).

References

[1] F. Wang, P. Liu, S. Zhao and B. Chen, Development of an unmanned helicopter for vertical replenishment, *Unmanned Syst.* **3**(1) (2015) 63–87.
[2] J. Scott and C. Scott, Drone delivery models for healthcare, in *Proc. 50th Hawaii Int. Conf. System Sciences*, 2017.
[3] E. Ackermann and M. Koziol, The blood is here: Zipline's medical delivery drones are changing the game in Rwanda, *IEEE Spectr.* **56**(5) (2019) 24–31.
[4] P. M. Kornatowski, A. Bhaskaran, G. M. Heit, S. Mintchev and D. Floreano, Last-centimeter personal drone delivery: Field deployment and user interaction, *IEEE Robot. Autom. Lett.* **3**(4) (2018) 3813–3820.
[5] S. S. Kannan and M. Byung-Cheol, Autonomous drone delivery to your door and yard, *2022 Int. Conf. Unmanned Aircraft Systems (ICUAS)*, Dubrovnik, 2022, pp. 452–461.
[6] N. Gyagenda, J. V. Hatilima, H. Roth and V. Zhmud, A review of GNSS-independent UAV navigation techniques, *Rob. Auton. Syst.* **152** (2022) 104069.
[7] G. Balamurugan, J. Valarmathi and V. P. S. Naidu, Survey on UAV navigation in GPS denied environments, *IEEE Int. Conf. Signal Processing, Communication, Power and Embedded System (SCOPES)*, 2016, pp. 198–204.
[8] P. Meincke, L. Asmer, L. Geike and H. Wiarda, Concepts for cargo ground handling of unmanned cargo aircrafts and their influence on the supply chain, *8th Int. Conf. Logistics, Informatics and Service Sciences (LISS)*, 2018, pp. 1–10.
[9] C. F. Liew and T. Yairi, Companion unmanned aerial vehicles: A survey, unpublished preprint, Tokyo, 2020.
[10] A. L. C. Doneda and J. C. de Oliveira, Helicopter visual signaling simulation: Integrating VR and ML into a low-cost solution to optimize Brazilian Navy training, *22nd Symp. Virtual and Augmented Reality (SVR)*, 2020, pp. 1–10.
[11] U. Navy, *Aircraft Signals NATOPS Manual*, 00-80T-113 edn., Washington, D.C., 1997.
[12] W. Yoo, E. Yu and J. Jung, Drone delivery: Factors affecting the public's attitude and intention to adopt, *Telemat. Inform.* **35**(6) (2018) 1687–1700.
[13] D. Tezza and M. Andujar, The state-of-the-art of human–drone interaction: A survey, *IEEE Access* **7** (2019) 167438–167454.
[14] N. Gio, R. Brisco and T. Vuletic, Control of a drone with body gestures, in *Proc. Design Society*, Vol. 1, 2021, pp. 761–770.
[15] Y. Yu, X. Wang, Z. Zhong and Y. Zhang, ROS-based UAV control using hand gesture recognition, *IEEE 29th Chinese Control and Decision Conf.*, Chongquing, China, 2017, pp. 761–770.

[16] P. Vogiatzidakis and P. Koutsabasis, Gesture elicitation studies for mid-air interaction: A review, *Multimodal Technologies and Interaction* **2**(4) (2018) 65–86.

[17] S. Villarreal-Narvaez, J. Vanderdonckt, R.-D. Vatavu and J. O. Wobbrock, A systematic review of gesture elicitation studies, *DIS '20: Designing Interactive Systems Conf. 2020*, New York City, 2020, pp. 855–872.

[18] A. K. Lampton, J. R. Gray and J. P. Miller, Formal evaluation of IMU-based gesture recognition for UAS aircraft carrier deck handling, in *AIAA Information Systems-AIAA Infotech@ Aerospace*, Kissimmee, 2018, pp. 75–97.

[19] K. Pfeil, S. L. Koh and J. LaViola, Exploring 3d gesture metaphors for interaction with unmanned aerial vehicles, in *Proc. 2013 Int. Conf. Intelligent User Interfaces*, 2013.

[20] E. Peshkova and M. Hitz, Coherence evaluation of input vocabularies to enhance usability and user experience, in *Proc. ACM SIGCHI Symp. Engineering Interactive Computing Systems*, 2017.

[21] E. Peshkova, M. Hitz and D. Ahlström, Exploring user-defined gestures and voice commands to control an unmanned aerial vehicle, in *Int. Conf. Intelligent Technologies for Interactive Entertainment*, Utrecht, 2016.

[22] C. C. Tsai, C. C. Kuo and Y. L. Chen, 3D hand gesture recognition for drone control in unity, in *IEEE 16th Int. Conf. Automation Science and Engineering (CASE)*, 2020, pp. 985–988.

[23] A. Schelle, *Visuelles Kommunikationssystem zur nonverbalen Mensch-UAV-Interaktion* (Athene Forschung Unibw, Neubiberg, 2022).

[24] M. Hitz, E. Königstorfer and E. Peshkova, Exploring cognitive load of single and mixed mental models gesture sets for UAV navigation, in *1st Int. Workshop on Human-Drone Interaction*, 2019, pp. 1–9.

[25] N. Wong and C. Gutwin, Where are you pointing? The accuracy of deictic pointing in CVEs, in *Proc. SIGCHI Conf. Human Factors in Computing Systems*, 2010, pp. 1029–1038.

[26] O. Herbort and W. Kunde, Spatial (mis-) interpretation of pointing gestures to distal referents, *J. Exp. Psychol. Hum. Percept. Perform.* **42**(1) (2016) 78–91.

[27] H. Cochet and J. Vauclair, Pointing gestures produced by toddlers from 15 to 30 months: Different functions, hand shapes and laterality patterns, *Infant Behav. Dev.* **33**(4) (2010) 431–441.

[28] S. Constantin, F. I. Eyiokur, D. Yaman, L. Bärmann and A. Waibel, Interactive multimodal robot dialog using pointing gesture recognition, in *European Conf. Computer Vision*, 2023, pp. 640–657.

[29] B. Gromov, G. Abbate, L. M. Gambardella and A. Giusti, Proximity human-robot interaction using pointing gestures and a wrist-mounted IMU, in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2019, pp. 8084–8091.

[30] B. Gromov, L. Gambardella and A. Giusti, Guiding quadrotor landing with pointing gestures, in *Int. Workshop on Human-Friendly Robotics*, Reggio Emilia, 2019, pp. 1–14.

[31] A. C. S. Medeiros, P. Ratsamee, Y. Uranishi, T. Mashita and H. Takemura, Human-drone interaction: Using pointing gesture to define a target object, in *HCII 2020: Human-Computer Interaction. Multimodal and Natural Interaction*, Part of the Lecture Notes in Computer Science book series (LNISA, Volume 12182), 2020, pp. 688–705.

[32] A. C. S. Medeiros, P. Ratsamee, J. Orlosky, Y. Uranishi, M. Higashida and H. Takemura, 3D pointing gestures as target selection tools: Guiding monocular UAVs during window selection in an outdoor environment, *Robomech. J.* **8**(1) (2021) 1–19.

[33] M. Brenner and P. Stütz, Towards gesture-based cooperation with cargo handling unmanned aerial vehicles: A conceptual approach, in *Sixth IEEE Int. Conf. Robotic Computing (IRC)*, Naples, 2022, pp. 8–17.

[34] A. Suarez, P. R. Soria, G. Heredia, B. C. Arrue and A. Ollero, Anthropomorphic, compliant and lightweight dual arm system for aerial manipulation, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Perth, 2017, pp. 558–574.

[35] L. Neumann and A. Vedaldi, Tiny people pose, in *Asian Conf. Computer Vision*, Vancouver, 2018, pp. 992–997.

[36] W. McNally, K. Vats, A. Wong and J. McPhee, Rethinking keypoint representations: Modeling keypoints and poses as objects for multi-person human pose estimation, *European Conference on Computer Vision*, 2022, Tel Aviv, pp. 37–54.

[37] C. K. Kang, *Development of a Peripheral-Central Vision System to Detect and Characterize Airborne Threats* (Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2020).

[38] J. Dichgans and T. Brandt, Visual-vestibular interaction: Effects on self-motion perception and postural control, in *Perception* (Springer, Berlin, Heidelberg, 1978), pp. 755–804.

[39] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, Simple online and realtime tracking, in *IEEE Int. Conf. Image Processing (ICIP)*, Phoenix, 2016, pp. 3464–3468.

[40] J. Akagi, T. D. Moon, X. Chen and C. Peterson, Gesture commands for controlling high-level UAV behavior, *SN Appl. Sci.* **3**(6) (2021) 23.

[41] H. Xia, M. Glueck, M. Annett, M. Wang and D. Wigdor, Iteratively designing gesture vocabularies: A survey and analysis of best practices in the HCI literature, *ACM Trans. Comput.-Hum. Interact.* **29**(4) (2022) 1–54.

[42] A. G. Perera, Y. Wei Law and J. Chahl, UAV-GESTURE: A dataset for UAV control and gesture recognition, in *Proc. European Conf. Computer Vision (ECCV) Workshops*, Munich, 2018, pp. 1–12.

[43] A. G. Perera, Y. W. Law and J. Chahl, Drone-action: An outdoor recorded drone video dataset for action recognition, *mdpi Drones* **3**(4) (2019) 82–98.

[44] T. Li, J. Liu, W. Zhang, Y. Ni, W. Wang and Z. Li, UAV-Human: A large benchmark for human behavior understanding with unmanned aerial vehicles, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Nashville, 2021, pp. 16266–16275.

[45] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan and C. L. Zitnick, Microsoft coco: Common objects in context, in *Eur. Conf. Computer Vision*, Zurich, 2014, pp. 740–755.

[46] M. Tan, R. Pang and Q. V. Le, Efficientdet: Scalable and efficient object detection, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle(Virtual), 2020, pp. 10781–10790.

[47] Z. Ge, S. Liu, F. Wang, Z. Li and J. Sun, Yolox: Exceeding yolo series in 2021, in preprint, arXiv:2107.08430, 2021.

[48] R. Lyu, NanoDet-Plus: Super fast and high accuracy lightweight anchor-free object detection model, 2021, https://github.com/RangiLyu/nanodet.

[49] C. Lyu, W. Zhang, H. Huang, Y. Zhou, Y. Wang, Y. Liu, S. Zhang and K. Chen, RTMDet: An empirical study of designing real-time object detectors, preprint, arXiv:2212.07784, 2022.

[50] C. Y. Wang, I. H. Yeh and H. Y. M. Liao, You only learn one representation: Unified network for multiple tasks, preprint, arXiv:2105.04206, 2021.

[51] C. Y. Wang, A. Bochkovskiy and H. Y. M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, In *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Vancouver, pp. 7464–7475.

[52] X. Xu, Y. Jiang, W. Chen, Y. Huang, Y. Zhang and X. Sun, DAMO-YOLO: A report on real-time object detection design, preprint, arXiv:2211.15444, 2022, pp. 1–10.

[53] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang and M. Grundmann, Blazepose: On-device real-time body pose tracking, preprint, arXiv:2006.10204, 2020, pp. 1–4.
[54] S. Kreiss, L. Bertoni and A. Alahi, Openpifpaf: Composite fields for semantic keypoint detection and spatio-temporal association, *IEEE Trans. Intell. Transp. Syst.* (2021).
[55] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang and L. Zhang, Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle (Virtual), 2020, pp. 5386–5395.
[56] H. S. Fang, S. Xie, Y. W. Tai and C. Lu, Rmpe: Regional multi-person pose estimation, in *Proc. IEEE Int. Conf. Computer Vision*, Venice, 2017, pp. 2334–2343.
[57] Y. Xu, J. Zhang, Q. Zhang and D. Tao, ViTPose+: Vision transformer foundation model for generic body pose estimation, preprint, arXiv:2212.04246, 2022.
[58] S. M. LaValle, A. Yershova, M. Katsev and M. Antonov, Head tracking for the Oculus Rift, in *IEEE Int. Conf. Robotics and Automation (ICRA)*, Hongkong, 2014, pp. 187–194.
[59] T. A. Jost, B. Nelson and J. Rylander, Quantitative analysis of the Oculus Rift S in controlled movement, *Disabil. Rehabil. Assist. Technol.* **16**(6) (2021) 632–636.
[60] J. Martinez, R. Hossain, J. Romero and J. J. Little, A simple yet effective baseline for 3d human pose estimation, in *Proc. IEEE Int. Conf. Computer Vision*, Venice, 2017, pp. 2640–2649.
[61] C. Ionescu, D. Papava, V. Olaru and C. Sminchisescu, Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments, *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(7) (2013) 1325–1339.
[62] I. Grishchenko, V. Bazarevsky, A. Zanfir, E. G. Bazavan, M. Zanfir, R. Yee, K. Raveendran, M. Zhdanovich, M. Grundmann and C. Sminchisescu, BlazePose GHUM holistic: Real-time 3D human landmarks and pose estimation, arXiv, 2022.
[63] G. J. Székely, M. L. Rizzo and N. K. Bakirov, Measuring and testing dependence by correlation of distances, *Ann. Statist.* **35**(6) (2007) 2769–2794.