RGB-D based Haptic Teleoperation of UAVs with Onboard Sensors: Development and Preliminary Results

Paolo Stegagno, Massimo Basile, Heinrich H. Bülthoff, and Antonio Franchi

The interested reader can find in [1] a more comprehensive description of the system presented in this extended abstract.

I. Introduction

Micro UAVs represent the ideal platform for many robotic task, such as exploration, mapping, and surveillance. Their unconstrained workspace and versatility allow to use them as flying sensors and actuators to reach and operate on places that are impossible to reach with ground mobile robots. On the other hand many real-world tasks require that one or more humans participate to the mission, e.g., in the case of search-and-rescue missions [2]. Recent works have proven that haptic feedback can be successfully used in order to increase the operator situational awareness (see, e.g., [3] and references therein) and thus have a positive impact on the human decisions.

For the reasons mentioned above, *haptic shared control of UAVs* is an emerging topic that attracted the attention of many research groups in the last years. Concerning the single-UAV case, the authors of [3] study the proper design of artificial force fields for the haptic cue when bilaterally teleoperating a UAV, while the design of an admittance control paradigm from the master side with position feedback is presented in [4]. Single-UAV teleoperation control based on the port-Hamiltonian approach is presented in [5] and extended in [6]. The design of a strategy to generate the haptic feedback as a virtual force based on both telemetric and optic flow sensors is presented in [7]. A novel force feedback user interface for mobile robotic vehicles with dynamics is shown in [8], and a novel force feedback algorithm that allows the user to feel the texture of the environment is presented in [9]. Concerning the more general and challenging, multi-UAV case, in [10], [11], [12], [13] the authors present an extensive framework to control a group of UAVs that can be interfaced with multiple operators by means of haptic devices, e.g., to control some generalized velocity of the group formation. In [14] the same authors have shown how that framework can be applied in the real world to perform teleoperation over intercontinental distances.

At the best of our knowledge none of the previous approaches dealing with haptic-teleoperation of UAVs have been experimentally proven on a platform that uses onboard sensors only. The majority of the works never addressed the problem in a real world scenario, either employing simulation or using external motion capture systems. In [9] the obstacles are detected through laser scanners, but the state for control purpose is still retrieved by an external camera system. The authors of [11] use onboard cameras to measure the relative bearings, but the velocities are also obtained through external motion capture.

The scope of this workshop contribution (which is a preliminary version of the work presented in [1]) is to present the preliminary results of an ongoing project aimed at the design of a platform for RGB-D based haptic teleoperation of UAVs. The proposed platform will be able to navigate in unknown indoor environment relying only on its own sensor perception, i.e., IMU and RGB-D measurements. This sensor equipment, is relatively more rich with respect to the standard IMU-camera integration setting. This choice brings several advantages but also some drawbacks. First, the depth measurements will be extremely useful because they will allow a metric estimation of the velocity. Monocular camera methods usually do not provide such information, or they need accelerated motions to retrieve this information from the IMU. In addition, the measurements coming from an RGB-D sensor can be easily used to perform reliable obstacle avoidance. On the contrary, RGB-D sensors as Microsoft Kinect or Asus Xtion are usually sensible to natural light, so the system is specifically designed for indoor navigation.

Velocity measurements of the quadrotor are obtained employing in our framework the Dense Visual Odometry (dvo) algorithm presented by Kerl et al. in [15], and released in the form of a ROS package. Preliminary experimental results shows the effectiveness of our approach.

The rest of the paper is organized as follows. In Section II we present our target hardware architecture, for which we plan to have all computation exploited onboard. However, at current stage of development some computation is still performed on an external PC, although the measurements used for estimation purpose all comes from onboard sensors. Section III introduces some notation and explains the working principle of the system. Section IV presents the current stage of development and preliminary results from one experiment, while Section V concludes the paper.

II. Hardware Architecture

The target quadrotor configuration of this work comprises the mechanical frame, actuators, microcontrollers, and inertial measurement unit (IMU) of the MK-Quadro, a relatively low cost platform developed by MikroKoper.[1] Its actuation system consists of four plastic propellers with a diameter of

P. Stegagno, M. Basile, H. H. Bülthoff and A. Franchi are with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany paolo.stegagno@tue.mpg.de, massimo.basile@tue.mpg.de hhb@tue.mpg.de, antonio.franchi@tue.mpg.de

[1] http://www.mikrokopter.de/

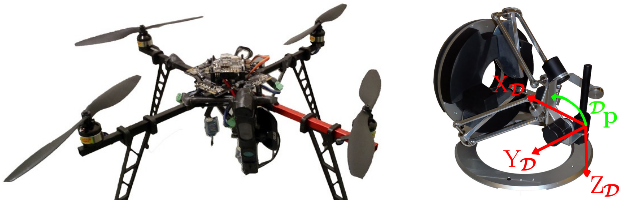

Fig. 1: Left: the quadrotor setup at date (i.e., without the Odroid board). Right: the haptic device and its frame of reference.

0.254 m, and a total span and weight of the frame of 0.5 m and 0.12 kg, respectively. The onboard electronics consists of:

i. a single board mini PC Odroid U2[2] for high level estimation, control, and interfacing purposes (this board is not installed yet but will be integrated soon);
ii. a low-level 8-bit Atmega1284p microcontroller connected to the mini-computer through two RS232 serial ports and a MAX232 converter.
iii. four brushless controllers connected to the low-level controller through a standard I2C bus;
iv. three 3D LIS344alh accelerometers and three ADXRS610 gyros, connected to the microcontroller;
v. a pressure sensor MPX4115A.

In addition, we have retrofitted the MK-Quadro frame with an Asus Xtion[3] RGB-D sensor, from now on referred to simply as 'camera', to obtain exteroceptive measurements of the environment. The camera is rigidly attached to the frame through three 5 mm diameter plastic bars, heading approximatively at 45° on the right of the quadrotor and tilted by approximatively 30° downward, vertically mounted to increase the field of view on the Z axis. In Fig. 1 (left) we report a picture of the whole system at date.

The whole system is powered by a 2600 mAh LiPo battery which guarantees an endurance of around 10 min of flight in normal regimes. The complete system has a weight of approximately 1.000 kg.

At current state of development, the Odroid board is still not integrated in the system. Its role is temporarily exploited by an external computer which communicates with the quadrotor through two XBee channels instead of wired serial connectors.

III. Flight Control and Teleoperation

Let be $\mathcal{W}$: $\{O_{\mathcal{W}}, X_{\mathcal{W}}, Y_{\mathcal{W}}, Z_{\mathcal{W}}\}$ the inertial (world) frame defined with the North-West-Up (NWU) convention, hence with $Z_{\mathcal{W}}$ pointing in the opposite direction of the gravity vector, and let be $\mathcal{Q}$: $\{O_{\mathcal{Q}}, X_{\mathcal{Q}}, Y_{\mathcal{Q}}, Z_{\mathcal{Q}}\}$ a frame attached to a representative point of the quadrotor (ideally its center of mass), which conforms to the North-East-Down (NED) convention as common in the aerospace field. In general, we will denote with ${}^{\mathcal{A}}\mathbf{p}_{\mathcal{B}}$ the position of the origin of a frame $\mathcal{B}$ in another frame $\mathcal{A}$ and with $\mathbf{R}_{\mathcal{B}}^{\mathcal{A}} \in SO(3)$ the rotation matrix expressing the orientation of the frame $\mathcal{B}$ in $\mathcal{A}$. With reference to the frames $\mathcal{W}$ and $\mathcal{B}$ we then define ${}^{\mathcal{W}}\mathbf{p}_{\mathcal{Q}} \in \mathbb{R}^3$ and $\mathbf{R}_{\mathcal{Q}}^{\mathcal{W}} \in SO(3)$. Finally, denote with ϕ, θ, ψ respectively the roll, pitch and yaw angles that represent the orientation of the quadrotor in $\mathcal{W}$, i.e., such that $\mathbf{R}_{\mathcal{Q}}^{\mathcal{W}} = \mathbf{R}_x(\pi)\mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$, where $\mathbf{R}_x(\cdot)$, $\mathbf{R}_y(\cdot)$, $\mathbf{R}_z(\cdot)$ represent the canonical rotation matrices about the axes X, Y, Z respectively.

It is convenient to introduce the (NED) horizontal frame $\mathcal{H}$: $\{O_{\mathcal{H}}, X_{\mathcal{H}}, Y_{\mathcal{H}}, Z_{\mathcal{H}}\}$ such that $O_{\mathcal{H}} \equiv O_{\mathcal{Q}}$ and $Z_{\mathcal{H}} \parallel -Z_{\mathcal{W}}$. Then, the rotation matrix between $\mathcal{H}$ and $\mathcal{Q}$ is $\mathbf{R}_{\mathcal{H}}^{\mathcal{Q}} = \mathbf{R}_z(\theta)\mathbf{R}_x(\phi)$. Finally, consider the camera frame $\mathcal{C}$: $\{O_{\mathcal{C}}, X_{\mathcal{C}}, Y_{\mathcal{C}}, Z_{\mathcal{C}}\}$. Since the camera is rigidly attached to the quadrotor, ${}^{\mathcal{Q}}\mathbf{p}_{\mathcal{C}}$ and $\mathbf{R}_{\mathcal{C}}^{\mathcal{Q}}$ are constant extrinsic parameters.

1) Human operator: The human operator interfaces with the system through an haptic device[4] shown in Fig. 1 (right). The device provides three translational actuated degrees of freedom (DOFs). Let be $\mathcal{D}$: $\{O_{\mathcal{D}}, X_{\mathcal{D}}, Y_{\mathcal{D}}, Z_{\mathcal{D}}\}$ the NED frame of reference whose origin is in the steady position of the end effector of the haptic device, then we define ${}^{\mathcal{D}}\mathbf{p} = (p_x\, p_y\, p_z)^T$ the configuration of the three translational DOFs of the haptic interface in $\mathcal{D}$. The commanded velocity for the quadrotor, expressed in $\mathcal{H}$, is then computed as:

$$\tilde{v}_x = k_v p_x \cos(\alpha) \tag{1}$$

$$\tilde{v}_y = k_v p_x \sin(\alpha) \tag{2}$$

$$\tilde{v}_z = -k_v p_z \tag{3}$$

where k_v is a positive gain and α is a parameter expressing the direction of the desired forward motion of the quadrotor. For safety reason, we want to force the quadrotor to move only in the direction in which the operator can see through the camera, hence α is selected as the yaw angle of the camera in $\mathcal{Q}$:

$$\alpha = \mathrm{atan2}(\mathrm{r}_{21}, \mathrm{r}_{11}) \tag{4}$$

where $\mathbf{R}_{\mathcal{C}}^{\mathcal{Q}} = [r_{ij}]_{i=1,\ldots,3,\, j=1,\ldots,3}$.

Notice that the commanded velocities are computed in the frame $\mathcal{H}$ instead of $\mathcal{Q}$ in order to let the command of the operator be independent from the roll and pitch motions that naturally arise when the quadrotor has to accelerate in the horizontal plane.

Finally the commanded yaw rate is obtained as

$$\tilde{\dot{\psi}} = -k_\psi p_y \tag{5}$$

where k_ψ is also a positive gain.

2) Obstacle avoidance and velocity tracker: We have implemented a simple obstacle detection and avoidance module in order to guarantee safe navigation and avoid contacts with the objects in the environment. In particular, the obstacles are detected by finding the local minima in the depth-component of the camera image. Then, a standard repulsive potential is applied to the point in order to avoid contact.

The resulting desired velocity is provided to the flight controller (referred to as 'velocity tracker' in the following) that uses also the current estimated velocity of the robot to compute the tracking error. The velocity tracker, described in [13], is a simple PD controller with gravity compensation computing the required roll and pitch angles

[2] http://www.hardkernel.com/

[3] http://www.asus.com/Multimedia/

[4] http://www.forcedimension.com/

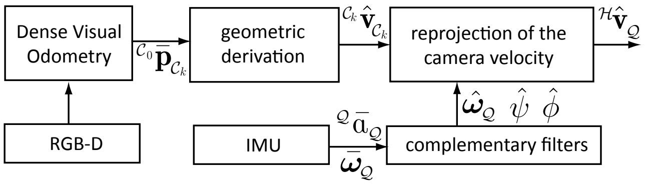
Dense Visual Odometry
${}^{\mathcal{C}_0}\bar{\mathbf{p}}_{\mathcal{C}_k}$
geometric derivation
${}^{\mathcal{C}_k}\hat{\mathbf{v}}_{\mathcal{C}_k}$
reprojection of the camera velocity
${}^{\mathcal{H}}\hat{\mathbf{v}}_{\mathcal{Q}}$
RGB-D
IMU
${}^{\mathcal{Q}}\bar{\mathrm{a}}_{\mathcal{Q}}$
$\bar{\boldsymbol{\omega}}_{\mathcal{Q}}$
complementary filters
$\hat{\boldsymbol{\omega}}_{\mathcal{Q}}$ $\hat{\psi}$ $\hat{\phi}$

Fig. 2: A block scheme representation of the estimation system.

that are needed to properly accelerate in order to track the velocity commanded by the human operator. Since it does not include an integral term, we expect the system to be affected by constant non-zero error in steady state, mainly due to non-perfect knowledge of the model. Given the nature of this work, the human operator will easily compensate for this error with the haptic device. Nevertheless, we are currently studying different controller strategies to eliminate this undesired behavior.

3) Velocity estimation: Figure 2 represents the implemented estimation system. The IMU measurements are used to compute estimates $\hat{\phi}_{\mathcal{Q}}$, $\hat{\theta}_{\mathcal{Q}}$ of the roll and pitch angles through a complementary filter (see [16], [17] for details). Similarly, a noise-free estimate ${}^{\mathcal{Q}}\hat{\omega}_{\mathcal{Q}}$ is obtained from the angular velocity measurement ${}^{\mathcal{Q}}\bar{\omega}_{\mathcal{Q}}$ from the gyros through a low-pass filter. At the generic time-step k the dvo[5] algorithm [15] provides ${}^{\mathcal{C}_0}\bar{\mathbf{p}}_{\mathcal{C}_k}$, $\bar{\mathbf{R}}^{\mathcal{C}_0}_{\mathcal{C}_k}$, which represent a drifting estimate of the position ${}^{\mathcal{C}_0}\mathbf{p}_{\mathcal{C}_k}$ and orientation $\mathbf{R}^{\mathcal{C}_0}_{\mathcal{C}_k}$ of the camera frame at time-step k, denoted with $\mathcal{C}_k$, w.r.t. the camera frame at the initial time-step $k = 0$, denoted with $\mathcal{C}_0$. Even though these estimates are drifting, denoting with $\mathcal{C}_{k-1}$ the camera frame at time $k-1$, it is possible to extract a velocity measurement using:

$$ {}^{\mathcal{C}_k}\mathbf{v}_{\mathcal{C}_k} = \mathbf{R}^{\mathcal{C}_k}_{\mathcal{C}_0}({}^{\mathcal{C}_0}\mathbf{p}_{\mathcal{C}_k} - {}^{\mathcal{C}_0}\mathbf{p}_{\mathcal{C}_{k-1}})/\Delta T \tag{6} $$

where ${}^{\mathcal{C}_k}\mathbf{v}_{\mathcal{C}_k}$ is the velocity of the origin $O_{\mathcal{C}_k}$ of the frame $\mathcal{C}_k$ expressed in $\mathcal{C}_k$, and ΔT is the elapsed time between time-steps $k-1$ and k. Since this estimate is considerably affected by numerical noise, a 1€-filter [18] is used on the visual odometry estimates to obtain a smoother estimate ${}^{\mathcal{C}_k}\hat{\mathbf{v}}_{\mathcal{C}_k}$ of the vehicle velocity. Since the velocity ${}^{\mathcal{C}}\mathbf{v}_{\mathcal{C}}$ of $O_{\mathcal{C}}$ in $\mathcal{C}$ can be expressed as

$$ {}^{\mathcal{C}}\mathbf{v}_{\mathcal{C}} = \mathbf{R}^{\mathcal{C}}_{\mathcal{Q}}\,{}^{\mathcal{Q}}\mathbf{v}_{\mathcal{C}} = \mathbf{R}^{\mathcal{C}}_{\mathcal{Q}}({}^{\mathcal{Q}}\mathbf{v}_{\mathcal{Q}} + {}^{\mathcal{Q}}\omega_{\mathcal{Q}} \times {}^{\mathcal{Q}}\mathbf{p}_{\mathcal{C}}) \tag{7} $$

we can compute an estimate of ${}^{\mathcal{Q}}\mathbf{v}_{\mathcal{Q}}$ at time-step k as

$$ {}^{\mathcal{Q}}\hat{\mathbf{v}}_{\mathcal{Q}} = \mathbf{R}^{\mathcal{Q}}_{\mathcal{C}}\,{}^{\mathcal{C}_k}\hat{\mathbf{v}}_{\mathcal{C}_k} - {}^{\mathcal{Q}}\hat{\omega}_{\mathcal{Q}} \times {}^{\mathcal{Q}}\mathbf{p}_{\mathcal{C}}. \tag{8} $$

Finally, given the estimates $\hat{\phi}_{\mathcal{Q}}$, $\hat{\theta}_{\mathcal{Q}}$ from the complementary filter, we have

$$ {}^{\mathcal{H}}\hat{\mathbf{v}}_{\mathcal{C}} = \hat{\mathbf{R}}^{\mathcal{H}}_{\mathcal{Q}}\,{}^{\mathcal{Q}}\hat{\mathbf{v}}_{\mathcal{Q}} \tag{9} $$

which can be then used in the velocity tracker.

IV. Implementation and Preliminary Results

We have conducted preliminary experiments in order to evaluate the performance of the implemented algorithms and obtain useful data for their improvement.

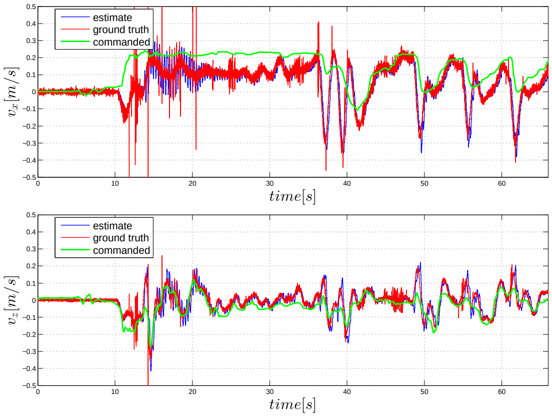

Fig. 3: Comparison between the estimated velocity (blue) with the velocity measured by an external Vicon system (red) and the commanded velocity (green). Spikes in the Vicon velocity are expected and due to outliers in the position measurements.

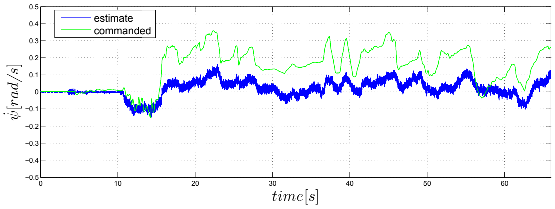

Fig. 4: Comparison between the measure yaw rate (blue) and the commanded yaw rate (green).

As mentioned in the previous section, our platform uses dvo as external tool in order to infer velocity measurements. However, the main framework in which the platform is developed is TeleKyb [19], a ROS-based project specifically designed for the design of applications on UAVs and oriented to multirobot execution.

At current stage of development not all computation is performed onboard, being the camera directly connected to an external PC which also hosts the execution of dvo and TeleKyb. The connection between the microcontroller and the PC is demanded to two pairs of XBee transmitters/receivers. Nevertheless, we plan to employ an Odroid U2 mini-board to replace the PC, and make the system really able to work in complete autonomy.

In the following, we present the results of a typical flight, for which the ground truth is given by an external camera system[6]. A video of the flight is also provided in the attached multimedia material and can be watched at antoniofranchi.com/videos/onboard_hapteleop.html.

In Fig. 3 we show the plots of the estimated (blue), real (red) and commanded (green) values of the three velocity components (expressed in $\mathcal{H}$). Commanded and measured yaw rate are shown in Fig. 4. All plots shows that the estimates are consistent with the real values of the velocities velocities are very similar to the real velocities. In addition, the quadrotor reproduces quite faithfully the commanded velocity, although a constant drift is present as theoretically predicted.

[5] https://github.com/tum-vision/dvo

[6] www.vicon.com

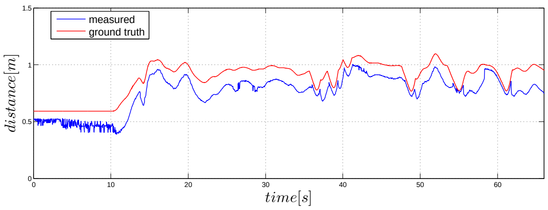

Fig. 5: Real (red) and measured (blue) distance between the quadrotor and the obstacle.

In order to test the obstacle avoidance module, we have tried to drive the quadrotor straight against an obstacle. We have performed this five times during the experiments, and in particular at times 38 s, 40 s, 50 s, 56 s and 62 s. In correspondence of those times, it is possible to recognize big spikes in the real velocities, which also significantly differs from commanded velocity due to the velocity correction term added by the obstacle avoidance algorithm.

In Fig. 5 we show the plot of the measured and real distances between the quadrotor and the surrounding obstacles. Note that as soon as the distance reaches the minimum admissible distance (0.7 m), the quadrotor is pushed back from the obstacle. This happens in correspondence of the five spikes in the plots of the velocities. In the first phase, the plot refers to the time before the take off when the obstacle avoidance is deactivated because of the vicinity with the ground.

V. Discussion and Ongoing Work

The ongoing project to perform teleoperation in unknown indoor environment is currently at a stage in which we are able to drive a quadrotor without the help of external navigation systems. However, not all computation is performed onboard, being the execution of dvo and TeleKyb software demanded to an external PC. The main drawback of the current system configuration is to have an USB cable connecting the onboard Xtion and the external PC, clearly limiting the motion of the quadrotor and disturbing its dynamics. In fact, the microcontroller alone is not able to acquire the output of the Xtion, nor to send it to the base station. Nevertheless, in our goal configuration we plan to connect the RGB-D sensor directly to the Odroid board, hence removing this issue.

Other improvements will consider different filtering strategies for the angular velocities and for the whole state. Once the platform is complete, we plan to employ it to perform teleoperation experiments over the internet, hence introducing significant delay on the commanded velocities.

References

[1] P. Stegagno, M. Basile, H. H. Bülthoff, and A. Franchi, “Vision-based autonomous control of a quadrotor UAV using an onboard RGB-D camera and its application to haptic teleoperation,” in *2nd IFAC Work. on Research, Education and Development of Unmanned Aerial Systems*, Compiegne, France, Nov. 2013.

[2] R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, and A. Erkmen, “Search and rescue robotics,” in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 1151–1173.

[3] T. M. Lam, H. W. Boschloo, M. Mulder, and M. M. V. Paassen, “Artificial force field for haptic feedback in UAV teleoperation,” *IEEE Trans. on Systems, Man, & Cybernetics. Part A: Systems & Humans*, vol. 39, no. 6, pp. 1316–1330, 2009.

[4] F. Schill, X. Hou, and R. Mahony, “Admittance mode framework for haptic teleoperation of hovering vehicles with unlimited workspace,” in *2010 Australasian Conf. on Robotics & Automation*, Brisbane, Australia, December 2010.

[5] S. Stramigioli, R. Mahony, and P. Corke, “A novel approach to haptic tele-operation of aerial robot vehicles,” in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 5302–5308.

[6] A. Y. Mersha, S. Stramigioli, and R. Carloni, “Switching-based mapping and control for haptic teleoperation of aerial robots,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 2012, pp. 2629–2634.

[7] H. Rifa, M. D. Hua, T. Hamel, and P. Morin, “Haptic-based bilateral teleoperation of underactuated unmanned aerial vehicles,” in *18th IFAC World Congress*, Milano, Italy, Aug. 2011, pp. 13 782–13 788.

[8] X. Hou, R. Mahony, and F. S. Schill, “Representation of vehicle dynamics in haptic teleoperation of aerial robots,” in *2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 1477–1483.

[9] S. Omari, M. D. Hua, G. J. J. Ducard, and T. Hamel, “Bilateral haptic teleoperation of VTOL UAVs,” in *2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 2385–2391.

[10] A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano, “Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs,” *IEEE Robotics & Automation Magazine, Special Issue on Aerial Robotics and the Quadrotor Platform*, vol. 19, no. 3, pp. 57–68, 2012.

[11] A. Franchi, C. Masone, V. Grabe, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano, “Modeling and control of UAV bearing-formations with bilateral high-level steering,” *The International Journal of Robotics Research, Special Issue on 3D Exploration, Mapping, and Surveillance*, vol. 31, no. 12, pp. 1504–1525, 2012.

[12] A. Franchi, C. Secchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano, “Bilateral teleoperation of groups of mobile robots with time-varying topology,” *IEEE Trans. on Robotics*, vol. 28, no. 5, pp. 1019–1033, 2012.

[13] D. J. Lee, A. Franchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano, “Semi-autonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles,” *IEEE/ASME Trans. on Mechatronics, Focused Section on Aerospace Mechatronics*, vol. 18, no. 4, pp. 1334–1345, 2013.

[14] M. Riedel, A. Franchi, H. H. Bülthoff, P. Robuffo Giordano, and H. I. Son, “Experiments on intercontinental haptic control of multiple UAVs,” in *12th Int. Conf. on Intelligent Autonomous Systems*, Jeju Island, Korea, Jun. 2012, pp. 227–238.

[15] C. Kerl, J. Sturm, and D. Cremers, “Robust odometry estimation for rgb-d cameras,” in *2013 IEEE Int. Conf. on Robotics and Automation*, May 2013.

[16] R. Mahony, T. Hamel, and J.-M. Pflimlin, “Nonlinear complementary filters on the special orthogonal group,” *IEEE Trans. on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[17] P. Martin and E. Salaün, “The true role of accelerometer feedback in quadrotor control,” in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 1623–1629.

[18] G. Casiez, N. Roussel, and D. Vogel, “1 euro filter: a simple speed-based low-pass filter for noisy input in interactive system,” in *SIGCHI Conference on Human Factors in Computing Systems*, Austin, Texas, May 2012, pp. 2527–2530.

[19] V. Grabe, M. Riedel, H. H. Bülthoff, P. Robuffo Giordano, and A. Franchi, “The TeleKyb framework for a modular and extendible ROS-based quadrotor control,” in *6th European Conference on Mobile Robots*, Barcelona, Spain, Sep. 2013.