Comparison of Tactile Signals for Collision Avoidance on Unmanned Aerial Vehicles

Stefan Spiss[1], Yeongmi Kim[2], Simon Haller[1], and Matthias Harders[1]

[1] Department of Computer Science, University of Innsbruck, Austria
[2] Department of Mechatronics, MCI, Austria
`stefan.spiss@student.uibk.ac.at`, `matthias.harders@uibk.ac.at`

Abstract. Our recent work focused on the development of intuitive user interfaces for the control of unmanned aerial vehicles, such as quadcopters. Next to intuitive gesture control, a key challenge with remotely operated quadcopters is the display of information about the aircraft surroundings. To this end, we examined the use of rendering tactile stimuli to warn about nearby obstacles. Directional information and distance is encoded via vibrotactile signals from rotating mass motors. Three different methods of delivering the tactile feedback were tested in a user study. Results show that even though participants guided the quadcopter through a maze by tactile stimuli alone, they were, on average, able to avoid full crashes. Further, we found that using sequential signals to indicate obstacles lead to significantly increased numbers of wall contacts.

Keywords: quadcopter, haptic, vibrotactile feedback, user interface

1 Introduction

Unmanned aerial vehicles (UAVs) are flight systems that do not carry a pilot, but are, in general, remotely controlled. Due to this, they are often smaller and more efficient. Also, no crew is put in possible danger during flight [1]. A key challenge for an operator is the loss of situational awareness of and knowledge about the surroundings of the aircraft. In most cases, UAVs are equipped with forward facing cameras to provide a video stream to the ground station. However, information about obstacles not in camera view is not readily available, which can be a problem in difficult environments, for instance in indoor rescue operations. Especially for quadcopters this is of concern due to their omnidirectional maneuverability.

In related work, the use of tactile feedback has been examined for cockpits [10] as well as for waypoint navigation of pedestrians, helicopter pilots, and boat drivers [8]. Using a vibrotactile waist belt led to performance improvement after only short familiarization. A similar application area is the use of tactile feedback for collision prevention and navigation in cars, e.g. [6, 3]. Moreover, related systems have also been employed as electrical travel aids for visually impaired persons to warn about obstacles [4, 5]. Generally, designs often take the form of torso [9] or belt-type displays [7]. The work closest to ours is by Brandt and Colton [2], who developed a collision avoidance system for quadcopters using a kinesthetic force-feedback device. In this paper vibrotactile signals are

Fig. 1: Quadcopter (left), gesture-based control, here with MYO bracelet, Oculus Rift (middle), Gazebo drone model, with virtual ultrasound sensors (right).

employed for providing directional cues in the horizontal plane. Also, optimal tactile feedback encoding patterns are explored in a user study.

2 System Overview

Quadcopter: We employ the *AR.Drone 2.0* from *Parrot*, shown in Fig. 1 (left). It comprises two cameras (one forward-, one downward-facing), an inertial measurement unit, a pressure sensor, and an ultrasound sensor. The video streams as well as the sensor data are sent over a WiFi network to a PC, using the Parrot SDK. To provide more intuitive control of the quadcopter, we built an intuitive gesture-controlled system, comprising of the MYO bracelet, the Leap Motion controller, and the Oculus Rift. Various head- and hand-gestures allowed the operator to fly the drone in an intuitive manner (see Fig. 1 (middle)). Redundancy in the control signals made the overall system more robust to noisy sensor readings. Nevertheless, the focus of this work is the tactile display of obstacle information, which will be outlined next.

Tactile Feedback Display: A vibratory signaling mechanism was developed to inform users about obstacles around the UAV. The display is comprised of four eccentric rotating mass motors. In the current state we only encode horizontal directions, i.e. front, back, left, right. The left and right tactors are placed outwards on the upper arms, the tactor in the back is located in the lower part of the back, and the one in front on the abdomen, below the solar plexus (Fig. 2 (right)). Each motor is placed inside a specially designed holder, shown in Fig. 2 (left); the latter being attached with Velcro to the body. The motors are controlled via pulse-width modulation using an *Atmega328p*. Tactile feedback is rendered according to the distance of obstacles detected in the surrounding of the quadcopter, such supporting the operator in collision awareness and avoidance. Various options for encoding this information via tactile signals exist, wherefore a user study was planned to evaluate the performance. Since an experimental metric would be the number of wall contacts as well as drone crashes, which would potentially damage the quadcopter, we opted for using a simulation of the quadcopter for the experiment. The underlying framework is introduced in the following.

Fig. 2: 3D model of tactor holder (left), user wearing attached tactors (right).

Simulation/Control Framework: For our project we employ the *Robot Operating System* ROS (`http://www.ros.org/about-ros`) – a framework that can be used for robot development. In order to control the AR.Drone with ROS, the *ardrone_autonomy* ROS driver (`http://wiki.ros.org/ardrone_autonomy`) was used, which is based on the official *AR-Drone SDK 2.0.1* (`http://ardrone2.parrot.com`). For the simulation of the quadcopter the *Gazebo framework* (`http://gazebosim.org`) was employed (see Fig. 1 (right) for a depiction), which is a robot simulator that supports ROS. The virtual drone representation is based on the ROS package *tum_simulator* (`http://wiki.ros.org/tum_simulator`). Note that for the user experiment virtual ultrasound sensors were attached to the outside of the quadcopter model to obtain accurate distance information in the horizontal flight plane. Further, a new ROS node was implemented to render tactile feedback according to detected obstacles.

3 Experiment

3.1 Methods

A user study was conducted to compare the performance of three different tactile obstacle encoding methods, in terms of efficiency and safety. The methods differ in the way information about obstacles is displayed to a participant. For the latter, obstacle distance is encoded by intensity, for three different range intervals, whereas a direction is encoded by activation of one of the tactors. Therefore, it is possible to give warnings for the four distances, left, right, front and back.

In the first method (*nearest only*), only information about the nearest obstacle is displayed. Thus, stimuli due to any other potentially close, but slightly further away obstacles are not generated. The second method (*all simultaneous*) presents feedback for any nearby detected obstacles, on all tactors simultaneously. In this paradigm, up to four tactors may be activated. In the third method (*all alternating*), all detected obstacles are rendered too, however, the corresponding tactile signals are presented sequentially. For the experiment, virtual environments with obstacles (i.e. three different maze classes, consisting of an original maze and its mirrored version) were designed in the Gazebo simulator, as illustrated in Fig. 3 (left). The goal was to navigate these mazes, from a starting position to the finish line. The starting point was always one meter away from the back wall and in the middle of the corridor. The maximum width of all corridors was 3 meters, whereas the minimum width was about 1.3 meters. The

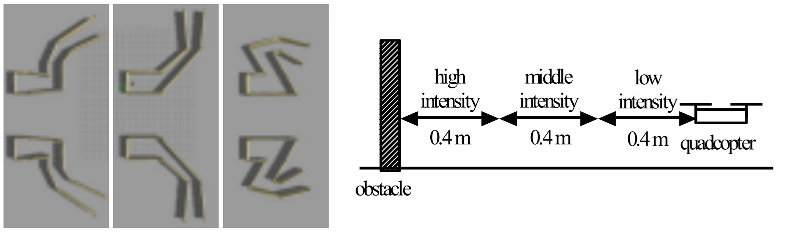

Fig. 3: Mazes used for tactile feedback experiment, each image showing one maze class (left), distance intervals used for different tactile intensities (right).

length from start to finish, along the centerline in the corridors, was 17 meters in all mazes. As goal line/plane that had to be crossed, the connection between the two ending wall edges was employed. Three intensities of vibratory feedback were applied for encoding equally sized distance intervals. High intensity (~ 2.0 g) was rendered, when an obstacle was closer than 0.4 meters, middle intensity (~ 1.3 g) for 0.4 – 0.8 meters, and low intensity (~ 0.8 g) for obstacles at 0.8 – 1.2 meters, respectively (see Fig. 3 (right)), and no signal for larger distances. Thus, when the quadcopter stayed on the centerline, in a corridor with maximum width, no feedback was presented. All tactile signals were displayed for 1/6 seconds, whereas the inter-stimulus interval was set to 1/12 seconds. Although the simulated quadcopter could be controlled via the developed gesture-based interface, we decided to use an XBox gamepad controller, to avoid distraction due to the interface. Moreover, the altitude of the quadcopter was fixed. Thus, it was not required to manually start or land the drone.

Ten male participants with a mean age of 23.5 years took part in the experiment. They were mainly recruited at the university and did not receive financial compensation. It was required that all subjects had prior experience with using the gamepad controller, to minimize possible learning effects. All participants were familiarized with the control and the environment in a training scene. During this, the mazes were visible so that subjects could get used to the quadcopter control, see what is happening when crashing, and also to get to know the different feedback methods. Following this, participants had to fly through test training mazes, at least two times per method, under the full experimental condition. For the experiment, subjects were asked to fly through mazes as fast and accurate as possible, while avoiding any crashes. During the experiment the mazes became invisible. However, as visual cues the prior trajectory, as well as the drone forward orientation was shown on a grid (top-down view) (see Figure 4). On the screen, a box with information on the trial was displayed, showing e.g. the remaining path distance to the goal. Moreover, various events in a trial were encoded with acoustic messages, e.g. the re/start of a trial or the crash of the drone. The tactile encoding methods and mazes were pseudo-randomly presented, for each method, each maze class three times, resulting in 27 trials per subject. As outcome measurements the number of wall contacts, the time

Current state:
Trial running.
Current method: 0
Trials left: 27
Distance to goal:

Fig. 4: Visual aid for participants during experiment.

of contact, the travelled distance, the number of crashes, the time of crashes, the trial completion time and the trajectory were captured. Note that real and simulated drones can contact walls without directly crashing. When a (simulated) crash occurs the quadcopter is reset to a prior position from where the trial can be continued. At the end of an experiment, subjects were asked about their preference of tactile encoding methods using a questionnaire.

3.2 Results

A repeated measures ANOVA revealed that there were significant differences in the number of contacts for the rendering methods ($F(2.0, 18.0) = 7.445,\ p = 0.004$), as well as for the duration of contact ($F(2.0, 18.0) = 8.601,\ p = 0.002$). Post-hoc tests employing Bonferroni correction showed that the *nearest only* method (5.36 ± 1.63 contacts per run) lead to significantly less contacts than the *all alternating* method (6.88 ± 1.54 contacts per run), ($p = 0.032$). In Fig. 5 (left) these results are depicted as boxplots. Further, there were significant differences in contact duration between *nearest only* ($p = 0.023$) as well as *all simultaneous* ($p = 0.027$) condition, with respect to the *all alternating* method (see Fig. 5 (right)). No significant results were found in other measurements, such as flight time, travelled distance, number of crashes, or time of crashes ($p > 0.05$). The mean number of crashes per flight, for all subjects and methods, was < 0.34. Moreover, a rank ordered preference for the three coding methods, as specified by the users, was examined via a Friedman test. Results show that there is a significant rank ordered preference ($\chi^2(2) = 8.667$, p=0.013). A post-hoc Nemenyi test indicates that the *all simultaneous* method was clearly preferred over the other two ($p = 0.01$ and $p = 0.037$).

4 Conclusion

In this paper an intuitive interface, comprising tactile feedback devices, for controlling quadcopters was introduced. In order to provide warnings about surrounding obstacles, three different tactile display methods were examined: feedback for only the nearest obstacle, for all detected obstacles simultaneously, and for all obstacles, but sequentially. The results of the user study indicate that the *all alternating* method provides lower performance in contact avoidance than the other two. In addition, the low mean number of crashes per flight of 0.34 shows that a majority of participants was able to control the flight of the quadcopter by tactile feedback only. Overall, it appears recommendable to employ

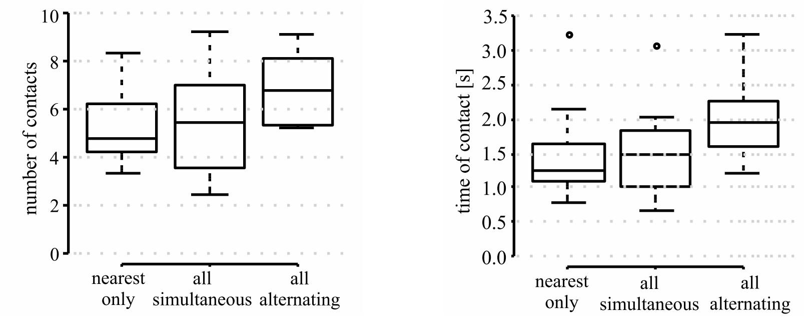

Fig. 5: Number of contacts (left), duration of contacts (right); with Tukey whiskers for data points at most $1.5 * IQR$ away from first or third quartile.

tactile feedback in addition to the video stream shown in the HMD in our setup. In the future, the tactile feedback should be extended to also present warnings about obstacles below and above the quadcopter. Moreover, the tactile feedback system should be tested in reality on an actual quadcopter and not only in a simulator. However, appropriate safety measures have to be taken. Finally, the combination with the HMD and the gesture-based control will be tested too.

References

1. Austin, R.: Unmanned aircraft systems: UAVS design, development and deployment, vol. 54. John Wiley & Sons (2011)
2. Brandt, A., Colton, M.: Haptic collision avoidance for a remotely operated quadrotor uav in indoor environments. In: Systems Man and Cybernetics. pp. 2724–2731 (2010)
3. Ho, C., Reed, N., Spence, C.: Assessing the effectiveness of intuitive vibrotactile warning signals in preventing front-to-rear-end collisions in a driving simulator. Accident Analysis & Prevention 38(5), 988–996 (2006)
4. Kim, Y., Harders, M., Gassert, R.: Identification of vibrotactile patterns encoding obstacle distance information. IEEE transactions on haptics 8(3), 298–305 (2015)
5. Pyun, R., Kim, Y., Wespe, P., Gassert, R., Schneller, S.: Advanced augmented white cane with obstacle height and distance feedback. In: Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on. pp. 1–6. IEEE (2013)
6. Scott, J., Gray, R.: A comparison of tactile, visual, and auditory warnings for rear-end collision prevention in simulated driving. Human Factors: The Journal of the Human Factors and Ergonomics Society 50(2), 264–275 (2008)
7. Tsukada, K., Yasumura, M.: Activebelt: Belt-type wearable tactile display for directional navigation. In: Intl. Conf. on Ubiquitous Computing. pp. 384–399 (2004)
8. Van Erp, J., Van Veen, H., Jansen, C., Dobbins, T.: Waypoint navigation with a vibrotactile waist belt. ACM Trans on Applied Perception 2(2), 106–117 (2005)
9. Van Erp, J.B.: Presenting directions with a vibrotactile torso display. Ergonomics 48(3), 302–313 (2005)
10. Van Veen, H.A., Van Erp, J.B.: Tactile information presentation in the cockpit. In: Haptic Human-Computer Interaction, pp. 174–181. Springer (2001)