Virtual Tourism with Drones: Experiments and Lag Compensation

David Mirk
University of Vienna
Entertainment Computing
Vienna, Austria
david@mirk.at

Helmut Hlavacs
University of Vienna
Entertainment Computing
Vienna, Austria
helmut.hlavacs@univie.ac.at

ABSTRACT

Tourism always involves physical movement between places, an activity that may be cumbersome, expensive, or even dangerous. Virtual tourism aims at reducing limitations by recreating real touristic venues in computers as 3D models. However, virtual tourism is always premade or prerecorded.

In our virtual tourism approach we use UAVs to fly around at the target venues, and send a video they record live to a tourist sitting at home, or walking in a treadmill. The video is presented in a VR visor, and head movements steer the orientation of the drone. This way tourists have the impression to really visit a remote site live, without any limitations. We present experimental results and analyze the overhead of compensating Internet network delay.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Tourism, UAVs, live streaming

1. INTRODUCTION

Virtual reality offers tourism many useful possibilities to create or extend virtual experiences that tourists may accept and use as partial alternatives for real visitation. The number of applications implementing these features will increase, as this technology continues to be developed and become more ready to go into production. Especially in some areas of tourism like marketing, entertainment or education, virtual reality will become more and more valuable.

The vision of this work is to allow users of a client software application to receive real-time video images of remote places of the world and watch them at home. As a special feature, and unlike to fixed stationary webcams, the user should be as free as possible to define where the "Virtual Eye" resides and what it is looking at. To enable a largely realistic experience for the user, to convey the feeling of actually being there, the motion control of the "Virtual Eye" should be realized by the detected changes of movement and viewing direction of the controller.

As a candidate for the "Virtual Eye" which almost meets all requirements and stands by a freely available software development kit for programming the control sequences and transcoding of the video image, as well as the possibility of the use of wireless network standards for the transmission of this data, the "Parrot ARDrone 2.0" was chosen.[1]

2. RELATED WORK

Web tourist information systems nowadays contains various sources of information for potential visitors, spanning from finding vacant hotel rooms, to maps, lists of outdoor activity possibilities, to fotos of the area. More and more tourist web systems offer live web cams showing the current weather situation. Virtual tourism is often associated with creating 3D models of real touristic sites [5]. Besides just rendering a 3D scene, virtual tourism systems can be combined with geo-information systems (GIS) [3].

To the best of our knowledge, flying drones (aka UAVs) so far have not been proposed as means for virtual tourism. Many scientific projects and papers working on (semi-) autonomous flying drones/UAV's or different types of multicopters are based on an indoor usage with optimal environment conditions, like zero wind speed or IR-markers and Laser-Triangulation for a more accurate self location in this nearly perfect room. An impressive demonstration of such a project is the "The Flying Machine Arena - Quadrocopter Ball Juggling" of the ETH Zürich in which two drones are playing tennis with each other.[2]

Navigating in outdoor areas is particularly difficult due to unpredictable weather conditions or suddenly appearing obstacles. One project taking up the challenge is the open source community project "Paparazzi"[3] encompassing an exceptionally powerful and versatile autopilot system for fixed-wing aircrafts as well as multicopters. This project was used e.g. in an work of TU Delft university of technology where student teams turned a ARDrone 2.0 in a fully autonomous drone.

DroNet'15, May 18, 2015, Florence, Italy.

DOI: http://dx.doi.org/10.1145/2750675.2750681.

[1] http://ardrone2.parrot.com/
[2] http://www.flyingmachinearena.org/
[3] http://wiki.paparazziuav.org/wiki/Main_Page

3. LIVE VIRTUAL TOURISM

Our approach is to use a cheap off-the-shelf drone (a Parrot ARDrone 2.0) as a flying camera, and be used in a live scenario by tourists to roam around remote areas and ideally be immersed in the application such that the tourist has the impression of really being there (Figure 1).

Figure 1: VTourist project components

To fulfill the task as complete as possible, we decided to choose a development pattern which divides the control application to be developed into three main components. Firstly, the server application, which is responsible for the direct control of the ARDrone, transmitting the calculated navigation vectors to the ARDrone. The second main component defines the graphical user client (client application), which processes and transmits the direct input to the ground station via TCP. Finally, the existing SDK for this purpose must be modified and extended in order to allow the user to use a wide a range of different control devices (e.g. "Oculus Rift").

The ARDrone 2.0 is very susceptible to outer influences (Figure 2). Moderate wind gusts deviate the drones very quickly from their original course which leads to a behavior where the electronics of the quadrocopter try to counteract this divergence with abrupt control maneuvers, what makes it difficult to provide a stable (vibration-free) video sequence to the user. Furthermore, the rigid mounting of the video

Figure 2: Outdoor tests with Oculus Rift

cameras leads to an undesirable side effect: basics quadrocopters control their flight direction by changing the speeds of their propellers and thus changes their attitude. At an acceleration to the front the drone tilts with the front camera toward the ground, on a flight back into sequence then turned toward the sky. To avoid this effect a modification with a steerable front camera on a ball joint would be necessary.

A further difficulty is the transfer delay of video and navigation data control packets that are partially summed by the different transmission method and encoding steps up to two seconds. To provide a realistic navigation experience this delay has to be minimized. This applies to both, on the one hand to the wireless link quality between the ARDrone and the respective ground station and on the other hand to reduce the amount of transmitted data to a minimum.

One of the main problems is the low maximum flight time of ARDrone due to the existing low battery capacity of 1100 - 1800 mAh, which remains still in a possible total flying weight of about 500 g. With optimized flight batteries maximum flight times of 18-20 minutes are currently possible. However, since a return to the ground station must always be possible, this leads to a maximum flight distance of about 9-10 minutes. To enable this project for a practicable use in the field of tourism, the basis weight of the ARDrone needs either to be reduced or higher capacitive LIPO batteries to be developed. Another solution would be a targeted intelligent management of several available drones, which at the required return time passes the connected client stream to the next available fully charged drone to guarantee a seamless and significantly longer flight experience for the user.

A practical test arrangement showed that it is not possible to give the user complete freedom of the control over his permit virtual eye. Collisions and obstacles could lead to crashes and destroy the ARDrone, and the maximal flying distance and time is limited due to the fact that a return flight with the existing battery capacity must still be possible. For this reason, a virtual route in the form of GPS waypoints must be predefined, on which the ARDrone travels within certain tolerances. As a result the user directly controls only the flight speed of the virtual tour, not the drone navigation directions. This however allows a completely free focus and control of the front camera orientation.

3.1 GPS

As mentioned before, a semi-autonomous flight of the ARDrone, without the possibility of direct view control and the direct intervention in the navigation, would arise various problems and difficulties. Due to the small narrow field of view of 91 degree and the rigidly mounted front camera a user-controlled direct navigation of the ARDrone would only be possible after several exercise passages. The specification of a flight course, which on the one hand is free of any obstacles and thus the risk of collisions sinks, and on the other hand provides optimized designed sight seeing courses, which lie within the flight range of the ARDrone and show the most important visual interests, allows the user to enjoy a satisfactory experience without the need of having knowledge of the actual flight behavior and functional ways of the quadrocopter. Due to the strong wind vulnerability of the ARDrone 2.0 a sudden departure from the set course by a gust of wind could not always be prevented. The usage of GPS aids the drone to automatically correct its position

based on the received GPS signals and leads the ARDrone back to its original course.

The ARDrone 2.0, with its rigidly mounted front camera, requires a continous correction of the GPS-based calculated navigation vector to allow users to rotate the entire drone on its z-axis while traveling to the next destination waypoint, in order to follow their head-tracked input commands. Before takeoff the drone's heading (delta angle value to absolute north (0 °) in degrees received from the digital compass) is stored and used to calculate the current offset of the drones actual orientation to this reference value.

Calculation of GPS based navigation vector from current position (lat1,lng1) to target waypoint (lat2,lng2):

$$\begin{aligned} \Delta(lng) &= lng_1\pi/180 - lng_2\pi/180 \\ y &= \sin(\Delta(lng))\cos(lat_2) \\ x &= \cos(lat_1\pi/180)\sin(lat_2\pi/180) - \\ &\quad \sin(lat_1\pi/180)\cos(lat_2\pi/180)\cos(\Delta(lng)) \\ \alpha &= 2\arctan(y/(\sqrt{x^2+y^2}+x))/(180\pi) \\ roll &= \cos(\alpha\pi/180) \\ pitch &= \sin(\alpha\pi/180) \end{aligned} \quad (1)$$

The angle α can further be represented as a vector from the current drone position to the next waypoint. This vector can be easily converted to usable roll and pitch values that can directly be used to control the ARDrone to fly into the desired direction.

3.2 Virtual Reality

The ARDrone 2.0 and the provided SDK already provides developers of control applications numerous possibilities for real-time symbol or character recognition. Necessary calculations are made directly on the ARDrone hardware/firmware and the results are processed to a client application. These features already are implemented for example in form of a virtual reality shooter: AR.PURSUIT (by Parrot SA). In addition to this onboard solution, it is also possible to implement more methods and algorithms for visual recognition into the client and/or server application in order to provide additional information, about objects or interactive elements such as video or audio recordings, during the local flight to the user's chosen output device.

Figure 3: Screen of rendered view via drone camera

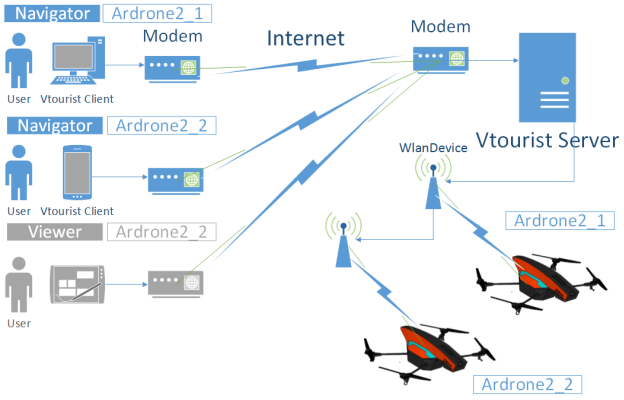

Figure 4: Overview of VTourist-project components

The Oculus Rift is a virtual reality headset, originally designed for immersive gaming.[4] To create a strong sense of immersion it is designed to almost fill the wearer's entire field of view and to completely block out the real world. The pincushioned image for each eye (640 × 800 resolution) shown in Figure 3 is corrected by lenses in the headset. The used combination of 3-axis gyros, accelerometers, and magnetometers make it capable of absolute (relative to earth) head orientation tracking without drift. These tracking features allow an accurate control of the direction of the drone mounted camera. In case of the rigid mounted camera of the ARDrone 2.0 every rotation of the users head leads to a chain of tilting actions to rotate the entire drone to the desired angle, which needs a continuous correction of the navigation vectors to stay on the predefined course.

3.3 Application

Figure 4 shows the basic structure of the virtual tourist application, which is divided into two main components, a server application and a client application. A centrally to all drones connected server accepts client requests, manages the connections between ARDrone and client and executes the desired control actions of the user.

The client application (Figure 5) allows a direct TCP connection to the server application and further the semi-autonome navigation of the ARDrone 2.0. A simple instrument display provides the user with accurate feedback on the current status of ARDrone 2.0, the sensors values and received control commands. Highly visible and accessible buttons for "emergency", "take off" or "land" enable the rapid intervention during flight without an additional input device.

3.4 Lag Compensation and Video Bitrate

An important aspect of virtual reality is the immediate response of the visual system to head movements, most importantly rotations, but also translations. Since we assume that the user is moving along a treadmill and thus essentially remains on his position, we only consider head rotations.

In our application the main obstacle is network lag. Every head rotation needs to be sent to the drone being far away, which in turn reacts and changes its bearing. The user will see the result after an entire round trip time has gone by,

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

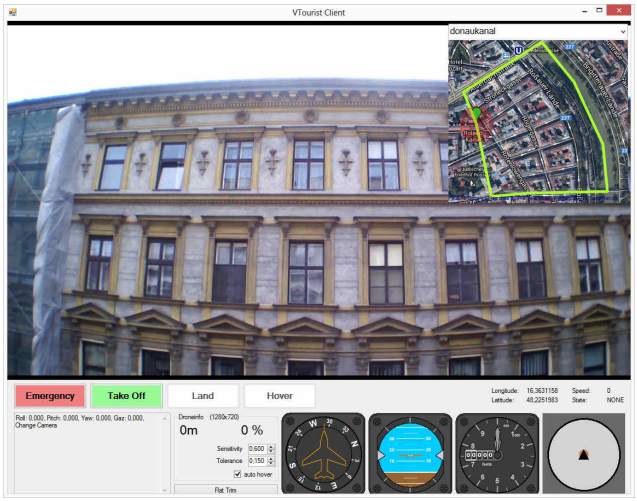

Figure 5: Screenshot of client application (desktop version)

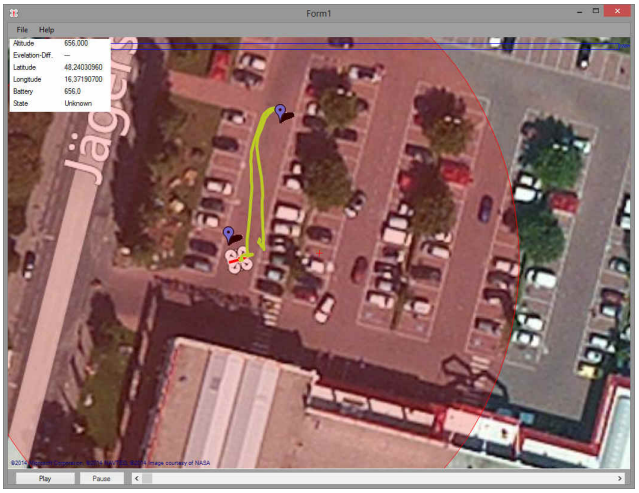

Figure 6: Plot of course differences in real environment

the RTT being thus the minimum experienced delay. Simple analysis shows that RTTs over the Internet can be quite substantial, for instance between Vienna and New York we measure an average of 240 ms of RTT, between Vienna and New Delhi we even measure an average RTT of 350 ms. Such lags are known to cause severe problems like fatigue, or nausea, and lags above 300 ms generally cause loss of immersion [17]. We therefore propose to compensate network lags by sending a larger image, and presenting only a part of it (a "window") to the viewer. We believe that network lags effectively can be hidden by immediately moving the shown window at the client side.

The following analysis takes into account only horizontal head rotations, an extension allowing also vertical head rotations can be defined similarly, but would increase the necessary bitrate. Assume a visor with 100 deg horizontal field of view (fov), and a user head mobility of 90 deg to left and right. Thus, the resulting possible field of view f of a person essentially moving forward on a treadmill is $f = 2 \times 90 + 100 = 280$ deg. The whole field could be recorded e.g. by two cameras with horizontal fovs of 140 deg each. Furthermore, the Oculus Rift DK2 shows a resolution of 960×1080 per eye. In the 2D case, this would entirely cover the visor fov of 100 deg. Assuming a similar resolution for our system, we would then require a video stream with horizontal resolution of $960 \times 280/100 = 2688$, and 1080 vertically, almost 3 Megapixels. Furthermore we use the fact that DVB with a resolution around 400 Kilopixels and encoder H.264 uses approximately 1.5 Mbit/s, and the new video codec High Efficiency Video Coding (HEVC) having twice the efficiency of H.264 would thus only need 0.75 Mbit/s for DVB. Thus, a 2D system using HEVC would need around $0.75 \times 30/4 \approx 5.6$ Mbit/s essentially without experiencing any lag. A 3D variant using two camera-pairs and Multi View Coding (MVC) would need roughly $5.6 \times 1.7 \approx 9.5$ Mbit/s with high quality and full lag compensation.

It is further be possible to reduce the transmitted video fov, such that it does not cover the whole 280 deg, but only a fov f between 100 and 280 deg. Movements of the head would shift the window at the client immediately, while instructing the drone to shift its sending window as well, the latter again being noticed by the user after one RTT.

In the minimum case $f = 100$ there would be no lag compensation, while in the maximum case $f = 280$ full compensation is attained. Any value $f \in [100, 280]$ in between can therefore compensate some but possibly not all head movements (and RTTs), but would require less bitrate. For instance, $f = 200$ deg would require bitrates of $5.6 \times 200/280 = 4$ Mbit/s for 2D, and $9.5 \times 200/280 \approx 6.8$ Mbit/s for stereo vision, while $f = 100$ requires only 2 Mbit/s and 3.4 Mbit/s respectively.

4. EXPERIMENTS

To prove the results given from the ARdrone 2.0 GPS receiver, especially the accuracy value, we carried out four different measurements of the received GPS packets while the drone stays still on the same point on ground (reference GPS values from GoogleMaps) over a short - 5 seconds and a long - 180 seconds period of time. In order to show how different environmental conditions like large buildings reflect or thick clouds absorb some of the received GPS signals, we have chosen one location representing some of the worst conditions, a small street (about 20 m width) surrounded by about 10-12 m high buildings and very cloudy weather conditions (Figure 9 and Figure 10) and the other one representing some of the best conditions, a wide open free parking area about 100 m×100 m size without any surrounding reflecting objects and partly cloudy conditions (Figure 7 and Figure 8). GPS Signals in this periods are collected in an interval of 10 ms.

4.1 GPS accuracy

In Figure 7 we can see that the mean of all received packets collected over 3 minutes has a distance to the reference point of 2.674 m which applies to the maximal specified possible resolution of 2 m of the GPS receiver. Some recorded packets are even hitting the reference point exactly. Comparing this result to Figure 8 showing the measurement at same location but only over a period of 5 s, resulting in 6.662 m distance, shows that an improvement of the accuracy e.g. by sampling and usage of statistic methods of the received packets would only be applicable at longer periods of time.

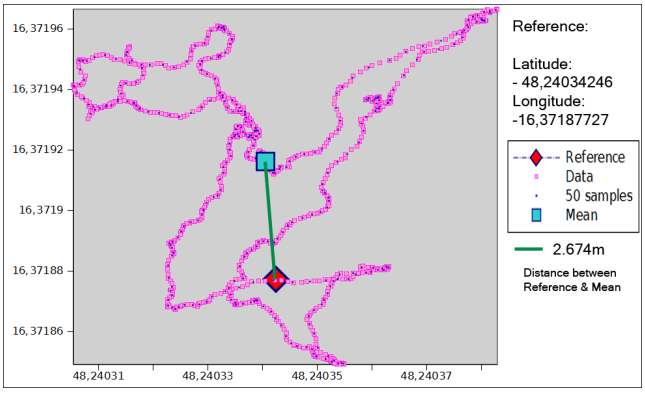

Figure 7: 2.674 m - 3 min measurement at an open place with light cloudy weather

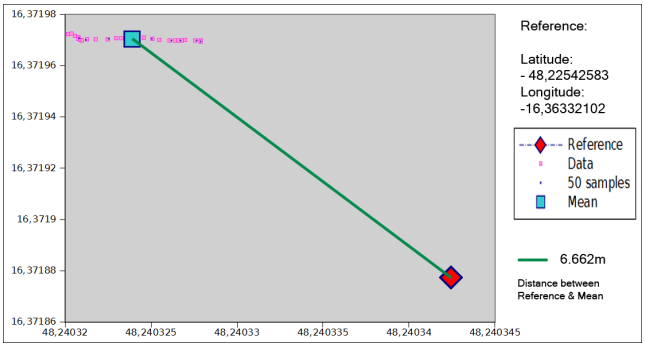

Figure 8: 6.662 m - 5 sec at an open place with light cloudy weather

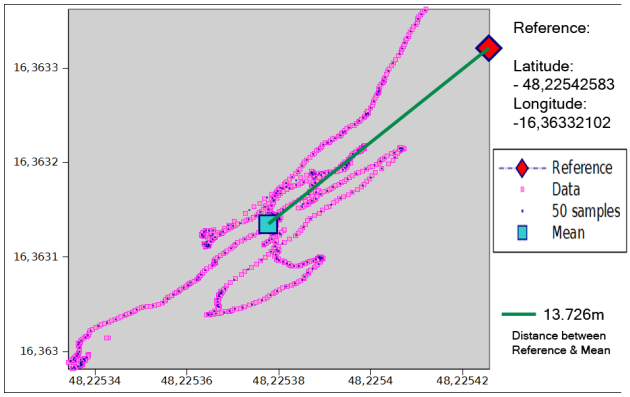

Figure 9: 13.726 m difference - 3 min measurement at a small street in city with cloudy weather

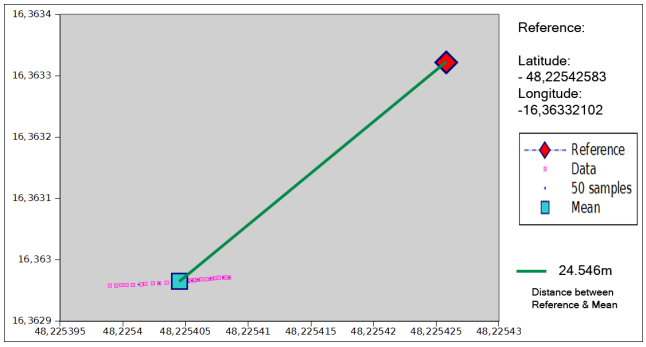

Figure 10: 24.546 m difference, 5 sec measurement at a small street in city with cloudy weather

These improvements can be used while the drone is hovering (during user stops and "look arounds") but not while navigates with regular horizontal speed to the next specified waypoint.

Figure 9 and Figure 10 show distances between the reference points and means of 13.726 m and 24.546 m which are absolutely not practicable for GPS based navigation of the drone. Considering the 20 m width of the street it is not even always possible to detect if the drones position is at a flyable (not inside of buildings) location.

These measurements and results shows that a lot of possible touristic locations may not be applicable for GPS-based navigation of flying UAVs. In order to allow an exact control of the drone and a smooth traveling between the specified waypoints good weather conditions and wide open areas with a minimum of large objects reflecting the GPS signals are required. In addition alternative implementations of location tracking e.g. DGPS (Differantial Global Position System) should be implemented to improve the accuracy.

5. CONCLUSIONS AND FUTURE WORK

Quadcopters are ideally suited to develop and test certain areas of the autonomous navigation of flying robots. The aim of this work is to develop a system that allows a user from anywhere on the world to control a ARDrone 2.0 via the Internet where the direct control of the direction of the quadrocopter is exclusively managed by the server application and only the alignment of the front camera is assumed by the user. Due to the difficulties and problems that occurred while developing and testing this solution with the Parrot ARDrone 2.0, which is a consumer market based solution for around 300$, lead us to the conclusion that for a practicable use of the Virtual Tourist in tourism, either the hardware/firmware of the current model needs to be optimized/redeveloped or a new quadrocopter model has to be chosen, which fulfills more of the given requirements.

We additionally analyse the overhead of Internet lag compensation. The resulting bitrates show that lag compensation is possible and theoretically could indeed render virtual tourism using drones feasible.

In the future, a possible alternative to the ARDrone could be the modular opensource DIY-Kit "ArduCopter" based on the APM 2.x autopilot. Because of the modular DIY system it is e.g. possible to use a wide range of available batteries with much higher capacities or to use the resulting higher possible additional weight that can be transported to attach professional video cameras (2D or 3D to extend the "Oculus Rift" support) on a panorama 2-axis mount at the bottom of the drone frame. An advantage of the rotateable bottom mounted camera over the rigid mounted front camera is the fact that no more movement of the drone frame is needed in order to change the actual field of view for the user, leading in a better "sightseeing flight" experience. In addition to mechanical image stabilization, methods and algorithms for software stabilization can be used to to compensate differences in camera images, caused e.g. by smaller wind shocks

or changes, by up to 15% and therefore offer the possibility to counteract visually within very short delays.

6. REFERENCES

[1] "AR.Drone open API platform" https://projects.ardrone.org/.

[2] Balas, C.: "Modelling and Linear Control of a Quadrotor." School of Engineering, Cranfield University, 2007.

[3] A 3D Geo Spatial Virtual Reality System for Virtual Tourism, V.F. Balogun, A.F. Thompson, O. A. Sarumi, The Pacific Journal of Science and Technology 11-2 (2010).

[4] Czyba, Roman.: "Attitude Stabilization of an Indoor Quadrotor." 2009

[5] R. Letellier, Virtual Reality - A New Tool for Sustainable Tourism and Cultural Heritage Sites Management, AIT, Bangkok, Thailand: February 1999; and Phimai, Thailand: February 1999.

[6] Mondragon, Ivn F, Miguel A Olivares-Mendez, Pascual Campoy, Carol Martinez, and Luis Mejias: "Unmanned aerial vehicles UAVs attitude, height, motion estimation and control using visual systems." Auton Robot, 2010.

[7] Daniel A. Guttentag: "Virtual reality: Applications and implications for tourism." Department of Geography and Environmental Management, MES Candidate in Tourism Policy and Planning, University of Waterloo, Waterloo, ON, Canada N2L 3G1

[8] Mark Müller, Sergei Lupashin and Raffaello D'Andrea: "Quardocopter Ball Juggling"

[9] Wenzel, Karl Engelbert, Andreas Masselli, and Andreas Zell: "Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle." J Intell Robot Syst 61. 2011.

[10] Alan C. Brooks: "Real-Time Digital Image Stabilization" (2003)

[11] Matti Niskanen, Olli Silven and Marius Tico of the Computer Vision Group at the University of Oulu: "Video Stabilization Performance Assessment"

[12] Marius Tico, Nokia Research Center Palo Alto, CA, USA: "Digital Image Stabilization"

[13] Alex Kushleyev, Daniel Mellinger, Vijay Kumar from GRASP Lab, University of Pennsylvania: "Towards A Swarm of Agile Micro Quadrotors"

[14] Jindrich Mrcek: "FSS Algorithm Adapted for Swarm Control of Unmanned Helicopters"

[15] Mohammad Shahnoor Islam Khan: "Implementation of Edge & Shape Detection Techniques and their Performance Evaluation" (2013) http://digitalcommons.ryerson.ca/cgi/viewcontent.cgi?article=2045&context=dissertations, Letzter Zugriff 04.09.2013

[16] TU Delft Minor Robotics: Quadrotor Group 1: "TU Delft - Search and Rescue with AR Drone 2" http://wiki.paparazziuav.org/wiki/TU_Delft_-_Search_and_Rescue_with_AR_Drone_2, Letzter Zugriff 04.09.2013

[17] G. Riva, B. Wiederhold, E. Molinari, Virtual Environments in Clinical Psychology and Neuroscience: Methods and Techniques in Advanced Patient-therapist Interaction, IOS Press, Amsterdam, 1998.