Immersive Visual Interaction with Autonomous Multi-Vehicle Systems

Ali Samini
ali.samini@liu.se
Linköping University
Norrköping, Sweden

Patric Ljung
patric.ljung@liu.se
Linköping University
Norrköping, Sweden

Figure 1: An overview of the system's desktop interface. Left: Control HUD, used to create and edit missions. Weather HUD, shows the local weather conditions. Right: Units HUD, showing List of all available units in the field with their information and settings. Middle: A mission path indicated with dashed blue lines between path points illustrated by red spheres. Also a virtual vertical pole with a blue sphere coupled with a sign showing its height which is used to create and edit path points.

ABSTRACT

With the emergence of multi-vehicular autonomous systems, such as AI controlled multiple fully autonomous vehicles, we need novel systems that provide tools for planning, executing, and reviewing of missions and keeping humans in the loop during all phases. We therefore present an immersive visualization system for interacting with these systems at a higher cognitive level than piloting of individual vehicles. Our system provides both desktop and VR modes for visual interaction with the robotic multi-vehicle AI system.

CCS CONCEPTS

• **Human-centered computing** → **Keyboards**; *User interface management systems*; **Virtual reality**; **Graphical user interfaces**; *Geographic visualization.*

KEYWORDS

visualisation, unmanned vehicle, virtual reality, UAV, USV

ACM Reference Format:
Ali Samini and Patric Ljung. 2021. Immersive Visual Interaction with Autonomous Multi-Vehicle Systems. In *27th ACM Symposium on Virtual Reality Software and Technology (VRST '21), December 8–10, 2021, Osaka, Japan.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3489849.3489917

VRST '21, December 8–10, 2021, Osaka, Japan

ACM ISBN 978-1-4503-9092-7/21/12.
https://doi.org/10.1145/3489849.3489917

1 INTRODUCTION

We present an immersive interactive visualization system for orchestrating unmanned areal, ground and sea vehicles (UAVs, UGVs, and USVs) for search-and-rescue missions with mixed-initiative capabilities. The system gives an operator the ability to create missions while also being able to follow and interact with ongoing missions. Our system supports both desktop and Virtual Reality (VR) interfaces and is developed using the Unreal Engine 4 (UE4) game engine. Figure 2 illustrates our system's main components including the visualization, planning, interaction, and communication parts. Arrows between components indicate communications and their directions. Our system works with a Delegation Framework (SymbiCloud HFKN Framework) [1], based on the Robotics Operating System (ROS) [5], for dynamic task allocations, which supports distributed data and knowledge collection, storage, and sharing among teams of collaborating units.

2 VISUALIZATION

The visual parts of the system consist of a 3D context of operations' environment, mission representations and multipurpose head-up display (HUD). For the environment a photogrammetry-based 3D model [3] around Gränsö Castle[1] has been created and integrated into the system. It is the location for outdoor tests and thus provides

[1]Gränsö Castle is a facility outside the town of Västervik in Sweden

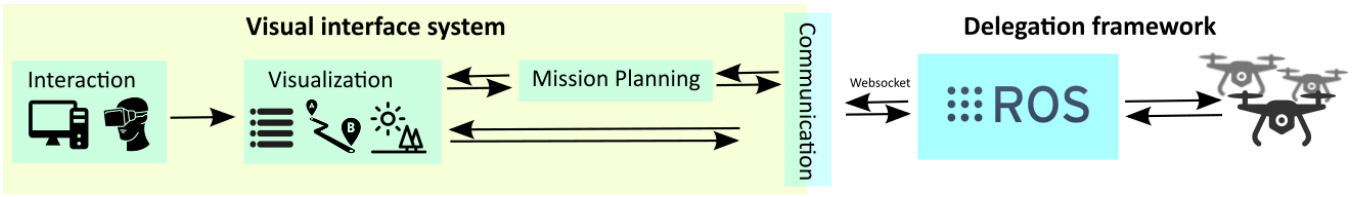

Figure 2: Schematic overview over our visual interface system and connectivity with the Delegation Framework.

Figure 3: The system's VR interface using a HTC View Pro Eye HMD and motion controllers.

a 3D context for the operators of our system. Figure 1 depicts a view of the castle's 3D model. The visualisation platform takes advantage of recent advancements in Unreal Engine, such as realistic atmospheric sky, day and night cycle, ocean water surface. Additional 3D models for UAVs, USVs, human actors, and battery station have also been added. Mission tasks are visualized with graphical elements such as path ways/points for unit movements and area indicators for scanning tasks. The multipurpose HUD consists of separate panels designed for specific display purposes: The *Units HUD* (Figure 1, right) shows a list of all available units in the operation's environment with their real-time info such as their delegation status, movement speed, and battery percentage. It is possible to select a unit and teleport to it. The camera stream from the units are also available. Ongoing missions for each unit can be stopped, paused or resumed, including a manual battery replacement order. The *Control HUD* (Figure 1, left) enables creating missions such as following a path, scanning, and leashing to another unit. Selecting each delegation opens a sub-panel that include its specific settings like unit speed or return home flag. The *Weather HUD* (Figure 1, second left) shows current weather information that can be useful for planning, specially for missions involving UAVs. There is also an interactive *mini-map* and *full map* that provide a 2D top-down view of the word while also showing the available units and ongoing mission visualizations.

3 INTERACTION

Interaction modes for both desktop and VR has been developed and designed as separate functionality layers to support switching between the two without changes in other parts of our system. Desktop interaction is controlled by mouse and keyboard where toggling between navigation and control modes changes their behaviours accordingly. With the aim to explore the utility of having a more immersive interface than desktop interface, we also developed a VR user interface. Figure 3 shows the system running with VR interaction mode. The units and controls menus are rendered attached to the left controller while they are interacted with a visual pointer from the right controller. Users can toggle between the menus, open and close the 2D map, execute or stop missions using pre-defined keys on the left controller while the right controller is used for navigation. Navigation is possible with both regular movement and teleportation.

4 MISSION PLANNING

The core feature of our system is about mission planning for the unmanned vehicles. The missions are designed, created and managed within our system before they are validated and passed to the delegation system for execution. Mission tasks include moving on a path, scanning an area, being leashed to a field operator, surveillance, battery change, and takeoff/land (for UAVs). Placing path points in the 3D environment is essential for creating either a path way or a scan area for a selected mission. For better perception of the height of the placed path points a virtual pole marks the ground underneath the path point while showing its height as depicted in the middle of Figure 1. The path points can be edited or removed after creation which leads to automatic re-generation of the created path or area. After mission planning the system sends the plan to the delegation server to begin the mission execution. Ongoing missions can be stopped, paused or resumed at any time. Real-time video streams from UAV cameras are available to watch within the system with both an automated predefined camera pose or manually controlling the gimbal camera. The human field operators are also tracked and visualized in the system where they can be used in scenarios such as leashing and battery change, which requires human assistance.

5 COMMUNICATION

Communication with the Delegation Framework is done over a Websocket interface to the ROS platform using Json formatted messages. The ROS communication uses a publish/subscribe protocol and a service call/response protocol. The latter is used to translate high-level mission plans in our system to Task Specification Trees [2] that is used by the Delegation Framework to execute and delegate tasks to all unmanned systems. Continuous updates such as position, pose, and other status is using the publish/subscribe protocol. Separate channels are used for video stream packets. Additionally, the Delegation Framework system is packaged into *Docker* [4] containers to allow for easy deployment.

6 CONCLUSIONS

Our system has been successful to deliver an immersive interactive visual environment and establishing a first approach to mission planning, execution and evaluation within such environment. The system is agnostic to whether the Delegation Framework is simulating scenarios or if it is executed in the operational real-world environment. The system provides a visual language to interact with and plan missions for systems of autonomous vehicles and allowing for mixed-initiative tasks. We are currently developing a touch-based interface, targeting both handheld devices and large touch tables. Using the VR, desktop, and touch interfaces we are also planning to study and evaluate the utility of all of these interface modalities for search-and-rescue missions with multiple autonomous vehicles.

ACKNOWLEDGMENTS

This work was supported by SSF grant RIT15-0097: *SymbiKBot: Robot-Assisted Hastily Formed Knowledge Networks.*

REFERENCES

[1] Cyrille Berger, Patrick Doherty, Piotr Rudol, and Mariusz Wzorek. 2021. Hastily Formed Knowledge Networks and Distributed Situation Awareness for Collaborative Robotics. arXiv:2103.14078 [cs.MA]

[2] Patrick Doherty, Fredrik Heintz, and David Landén. 2010. A Distributed Task Specification Language for Mixed-Initiative Delegation. In *Principles and Practice of Multi-Agent Systems, 13th Internation Conference* (Kolkata, India) *(PRIMA'10)*. Springer-Verlag, Berlin, Heidelberg, 42–57. https://doi.org/10.1007/978-3-642-25920-3_4

[3] Guang-kun Li, Fan Gao, and Zhi-gang Wang. 2011. A photogrammetry-based system for 3D surface reconstruction of prosthetics and orthotics. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 8459–8462. https://doi.org/10.1109/IEMBS.2011.6092087

[4] Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux journal* 2014, 239 (2014), 2.

[5] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. 2009. ROS: an open-source Robot Operating System. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA), Workshop on Open Source Robotics*. Kobe, Japan.