A Semantics of Spatial Expressions for interacting with unmanned aerial vehicles

Lucas C. F. Domingos
Centro Universitário FEI,
SBC, São Paulo, Brazil
ldomingos@fei.edu.br

Paulo E. Santos
College of Science and Engineering
Flinders University
Adelaide - South Australia
paulo.santos@flinders.edu.au

Abstract

This paper describes an investigation of establishing communication between a quadrotor and a human through qualitative spatial relations allied with an off-the-shelf speech recognition software. The quadrotor used in this research was equipped with GPS, IMU sensors, and radio communication, which was connected to a computer acting as a ground station. The ground station was implemented to interpret the received commands, correctly providing answers to the user according to an underlying qualitative reasoning formalism. The results obtained during the tests show that the error rate related to the answers given by this system was less than five per cent for vertical and radial dimensions. In contrast, commands related to the horizontal extent had an error rate of almost ten per cent.

1 Introduction

Unmanned aerial vehicles (UAV) have been gaining popularity in recent years due to their potential for novel applications (Shakhatreh et al., 2019). One of the most well-known types of UAVs is the quadrotor, owing to their fair cost-benefit and a large number of off-the-shelf programming tools available for application development. Some potential current and future activities involving UAVs include, for instance, mapping large areas (Achtelik et al., 2009), recording movie scenes (Fleureau et al., 2016), and search and rescue missions (Malfaz and Salichs, 2004).

One of the challenges for achieving a large-scale use of UAV applications, however, comes from the need to make more natural the way humans interact with such vehicles, especially for the non-specialised public (Franchi et al., 2012). This issue justified the development of an area of research known as human-robot interaction (HRI). HRI aims to develop strategies for facilitating the interaction with robotic agents in various situations, such as teaching children, rehabilitation, housework and many others. However, HRI is still to be considered in the context of UAV applications. Nevertheless, the most direct way to achieve a high level of communication and understanding between robots and humans is the vocal commands usage to transmit and answer the commands between these agents.

When two people want to talk to each other in everyday situations, they rarely use quantitative information, especially when talking about space and its relations (Aoyama and Shimomura, 2005). For example, when we say to a child to catch something at a table we do not tell the distance in meters or the relative altitude, we just give the basic qualitative information, like if it's close or far, under the table or on it. This observation motivated the present investigation, which aim is to develop new methods of human-robot communication using qualitative information. In general terms, this work aims to bridge the gap in the communication between a human and a quadrotor using speech recognition and a qualitative way of interpreting commands. Ideas from qualitative spatial reasoning (Cohn and Renz, 2008) will be used to provide the basis for this communication.

2 Related Work

The research reported in this paper is related to Qualitative Spatial Reasoning (QSR) (Cohn and Renz, 2008), which is a subfield of Knowledge Representation in AI that aims at the formalisation of spatial knowledge and the development of reasoning methods about this knowledge. In HRI, QSR ideas used a probabilistic model of interactions (Dondrup et al., 2015) based on the Qualitative Trajectory Calculus (QTC) (Van de Weghe et al., 2005). In that work, the robotic agent had to interpret the space around it while making decisions to avoid collisions and interacting with a human operator using qualitative information. More recently, (Perico et al., 2021) presents a multi-robot localisation system based on qualitative spatial information where a sensory-deprived robot was guided to a goal location by other robots by passing high-level spatial commands. Although no human interaction was considered in (Perico et al., 2021), the system presented would be suitable for achieving a human level of representing spatial concepts, as it has the right combination of qualitative representation with probabilistic localisation.

Another relevant work where QTC relations were used to enable autonomous agents to make decisions and predict actions from other agents by using just qualitative information, was presented in (Moratz and Ragni, 2008). Communication using spatial expressions was also considered with the introduction of a new formalism about qualitative location, named Qualitative Ego-Sphere (Rodrigues et al., 2016). The parameters of this formalism were obtained from human trials, and the resulting model was applied to two distinct situations: the first involved the information exchange between two robotic agents, and the second involved the interaction between a robotic agent and a human. As we shall see further in this paper, the Qualitative Ego-Sphere model was used as the basis for the research reported here; however, this idea was extended in the present work by assuming a flying robot as the robotic agent interacting with a human.

Much work has been done recently on deep learning for speech recognition using large language models, such as BERT (Devlin et al., 2019), GPT (Brown et al., 2020) among others (Sun et al., 2022). Although these models show great accuracy in actual language interactions, the semantics of their language constructs is unclear. In contrast, there is a growing interest in the development of formal semantics for spatial expressions, providing a rigorous account for verbal communication (Kelleher and Dobnik, 2022; Richard-Bollans et al., 2020; Rodrigues et al., 2020). This work presents a preliminary application of these ideas in the context of human-robot interaction.

3 Background

This work considers a discretisation of the space around an agent defining the Qualitative Ego-Sphere formalism to obtain successful communication using qualitative information, as presented below.

3.1 Qualitative Ego-Sphere

The qualitative Ego-Sphere (Rodrigues et al., 2016) is a qualitative spatial formalism based on a spherical shape to define the relative position of several points concerning the centre of a virtual sphere around an object, which could be an observer. This defines a qualitative egocentric reference system that can be considered a tridimensional generalisation of the Ternary Point Calculus (Moratz and Ragni, 2008).

To define the Ego-Sphere, the space around the agent (point of view v) is considered a discretised sphere. The first point of analysis is the discretisation of the radial distance relative to the point of reference v, which can be understood as defining regions of space that are referred to as *at*, *near* or *far* (cf. Figure 1). The category *at* is defined as the closest distance to the point of reference, considered as the minimum distance to avoid collisions; *near* is considered as the distance that can be reached by the agent in a short time if the speed is maintained constant, that is, it is a region that is close enough to the agent to be considered its close vicinity; *far* is defined as everything that is at a distance where the agent takes a longer time to reach. These three relations are similar to the human way of conceptualising space and can be understood as part of our commonsense knowledge.

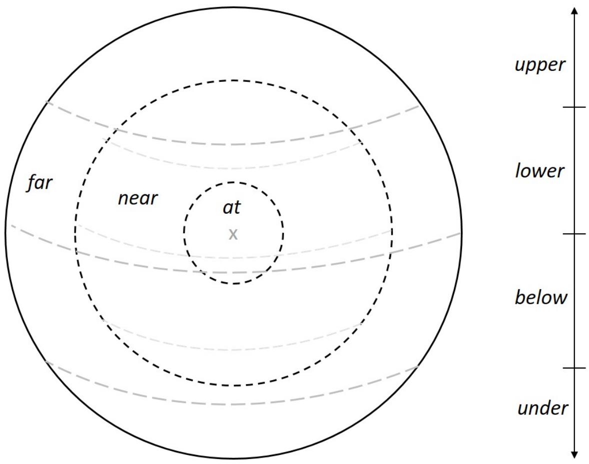

Figure 1: Ego-Sphere related at the point of view v

The second analysis area is divided into four different components called *upper*, *lower*, *below* and *under*, as shown in Figure 1. These components represent the altitude on the vertical level, and they depend directly on the dimensions of the agent: the greater the dimensions of the agent, the greater the distance between the division ranges.

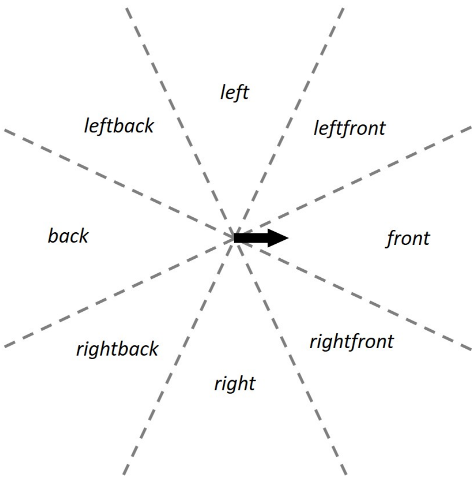

Figure 2: Relative positions regarded to a point of view v

The final subdivision considered in this work is a horizontal representation of directions, which has at its basis the 8-Star Calculus (Renz et al., 2004). This discretisation contains eight distinct regions, that are called *front*, *left-front*, *left*, *left-back, back, right-back, right* and *right-front*, respectively abbreviated as *f, lf, l, lb, b, rb, r* and *rf*. These relations are depicted in Figure 2. Figure 3 shows the resulting model combining all of these relations.

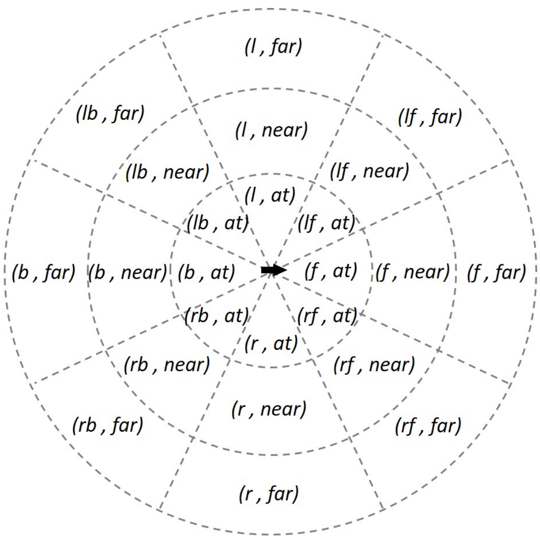

Figure 3: Horizontal relations with Ego-Sphere

An example of the use of the Ego-Sphere resides in the normal actions of daily life, such as the act of a child searching for some object in a dark room. The coordinates to find the object could be given as: "The object is *near*, at your *left side* and *above* you". A child can easily find that object if she follows the commands correctly; the same is expected from a robotic agent when high-level locations, such as "*Near. Left. Upper*", are given.

4 Experimental Setup

The quadrotor used in this research was an Arducopter with an APM 1-2560 board, an IMU board, a radio receiver, two XBee's for the telemetry, and a GPS. The dimension of the vehicle is 64x64x18 cm.

To use the concept of Ego-Sphere applied to this quadrotor, we have to consider that the UAV was the point of view v. The dimensions of the quadrotor were very important for this development, as well as its actuation area, in order to define the qualitative model. In this context, the region *at* was considered as a 1m radius centred at v, because it is the shortest distance to avoid a collision that can be perceived by the UAV GPS system. Similarly, *near* and *far* were considered as 5m and 10m respectively. The vertical discretisations of the sphere, *upper, lower, below* and *under*, received the values 5m, 2m, -2m, -5m, respectively. A new parameter was added to the latter category, the *same* command, as it was necessary to command the robot to stay at its current location. The final category, the horizontal location, was divided equally on the trigonometric circle so that each command would have 45 between adjacent regions in the circle.

The first step of this research was to control the UAV, and for that we used a range of existing software, such as the Ground Control Station (*GCS*), Radio transmitters, mobile apps, and others. We chose a GCS to control our system, as it can be installed on any computer and can have a wide range of peripherals attached to the system. The software used was the Ardupilot Mission Planner[1], and it has all the functions and tools needed for controlling this kind of drone. This software has a very large range of applications, such as support for the autonomous mission, control of all the optional hardware for this model and visual control of the basic functionality needed to fly. Using this software it was not difficult to find the appropriate function to have direct control from the computer.

All the commands sent to the quadrotor were in a specific message type using MAVLink (Meier et al., 2011). The meaning of the prefix MAV is *Micro Air Vehicle*, which is a common element of a large

[1] https://ardupilot.org/planner/

variety of UAV applications. MAVLink protocol is a library used in several programming languages that contain functions to translate and send messages between the vehicle and the control station, in this case, a computer. Thus, this library was a tool needed for the GCS, bringing standard protocol and portability to our code, making it possible to use the same code on other GCS or other software that used the same protocol. The idea of code portability was the main reason to use this protocol. This usage also made it possible to send flying commands using XBee, a radio transmitter/receiver module integrated into the quadrotor attached to the base system.

The first attempt at developing the interface between the control station and the quadrotor was to emulate radio signals from the computer and send them as normal commands by the radio transmitter. However, that was not a good approach, as those radio commands were very specific and did not have any type of support for autonomous flight. An alternative was to control the drone from the specific autonomous commands available in the Mission Planner, and using these commands implies using the full platform of the ground station and all the functionalities present in this software also. To accomplish this task, the Flight Plan tab of the Mission Planner was modified to work with direct commands and not a specific mission, as originally designed. For that, it was necessary to include equations and functions about latitude and longitude coordinates. Equations 1 and 2 describe how this information was used to determine the future trajectory points.

$$Lat_{end} = \sin^{-1}(\sin(Lat_{start}) \times \cos\delta + \cos(Lat_{start}) \times \sin\delta \times \cos\theta) \quad (1)$$

$$Long_{end} = atan2(\sin\theta \times \sin\delta \times \cos(Lat_{start}), \cos\delta - \sin(Lat_{start} \times \sin(Lat_{end})) + Long_{start} \quad (2)$$

In the equations above:

- Lat_{start} is the initial latitude of the drone;
- Lat_{end} is the destination point latitude;
- Long_{start} is the initial longitude of the drone;
- Long_{end} is the destination longitude of the drone;
- δ is the angular distance d/R;
- R is the Radius of the earth;
- θ is the bearing (clockwise from north).

Altitude commands were sent directly by the MAVLink protocol, using the data from the IMU board, which contains a barometer, an accelerometer and a gyroscope; however, distance and direction were sent by latitude and longitude. After receiving GPS signals and calculating the future point, we created the functions to control the Ego-Sphere commands, such as *left, upper* and *near*.

Being able to control the drone directly over the control station, without the radio controller, allowed the implementation of the voice recognition system. We adopted the Microsoft Speech library from Visual Studio (Johnson, 2012) as the basis for the voice recognition system, as this library allowed the processing of voice commands directly from the control station and sending commands to the drone without using the onboard computer in the drone.

The speech recognition module worked well with our functions, serving as the interface between human users and the ground station. The commands listed in the Table 1 were all the basic commands used to control the drone. Besides the basic commands, we have developed the Ego-Sphere commands, as explained above.

Table 1: The basic commands of the voice recognition

Command	***Description***
OK plane	Start the Ego-Sphere commands
Start the engines	Turn on the motors
Stop	Turn off the motors
Take off	Soars to a height of five meters
Down	Land at the same position
Stabilize	Starts the stabilize mode and change the control to the radio controller
Return to Launch	Returns to the initial position and land

All the voice commands used in this work have been adapted to reduce the error rate of recognition.

The voice recognition had great precision without background noise, especially because every word processed by the system was approximated by a previously defined word in the user-defined vocabulary. However, if a spurious sound is similar to one of these words, it can be misclassified as a valid command. To avoid such recognition problems, we configured the library with a confidence precision of 85%, which reduced ambiguity drastically.

A *grammar* class was used as a reference to the voice recognition module so that the application could use the language constraints in the recognition, which increased the hit rate of recognising commands. Four grammatical rules were defined containing different commands categories: the first contained all the basic commands (*Ok plane, start the engines, stop* etc); the second had the horizontal dimension of Ego-Sphere (*left, front, right* etc); the third was defined with vertical dimensions of Ego-Sphere (*upper, lower, under* and *below*) and; the last had the radial distance defined by Ego-Sphere (*at, near* and *far*). For our system to change the grammar at the appropriate time, we needed to establish an order of commands. The order was to call the horizontal references first, then the vertical and finally the distance, all according to the Ego-Sphere definitions. This order is described at Figure 4.

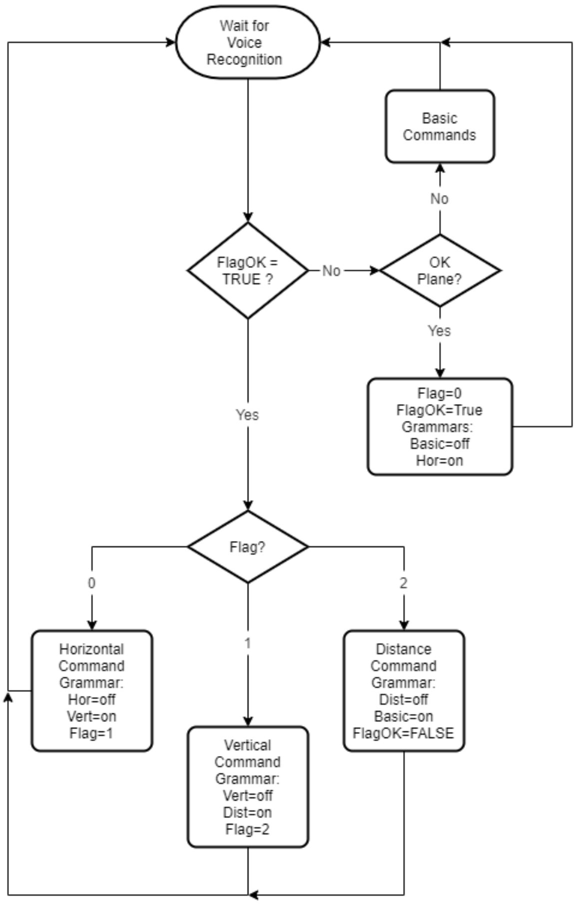

Figure 4: Flowchart of grammar interpretation

5 Results

A flying test was necessary to check if the recognition accuracy would satisfy the project goals and if the tests were consistent when flying with the radio by using direct commands. We found that the autonomous flight had certain issues, such as the stabilisation that was not precise and problems with the altitude holding. So, although the code was entirely developed for the physical platform, the evaluation of the system developed was conducted in a simulated engine, called Flight Gear (Perry, 2004). The usage of Flight Gear gave us a virtual ambient that emulates real flight, so every sensor data was received with precision and every command was sent with minimum delay compared with a real, non-simulated, flight.

The first test was conducted considering the basic commands, whereby we observed that commands with similar sounds, such as *arm* and *disarm* were not possible to be used on this application, because the similarity between these two words generated ambiguity in their recognition. So the major portion of the commands had to be changed to other words, which did not generate any kind of ambiguity. The final version of the basic commands was listed on Table 1.

Testing Ego-Sphere commands took longer than testing the basic commands, due to the complexity of the theory and the number of different words to be recognised. Subsequently, we divided the Ego-Sphere into two identical parts considering its symmetry, passing by the centre in a vertical cut, thus dividing the left side from the right side. For ease, just the left side was used in the experimental evaluation.

After dividing the sphere, four test sessions were executed with thirty complete commands in each one of them, approximately. One complete command was composed of three Ego-Sphere commands, one of each dimension. Every command given to the system was analysed according to the theory described before. Thus, to consider the command successful, we needed to analyse each one of the categories separately. To neutralise the influence of the different combinations of words, every session had the same list of commands executed in a different order, embracing a large range of possibilities. The tests were made on different days with different noise rates, with about 75dB of noise, composed of background voices and ambient sounds. That was made to maintain a realistic noise rate, representing the behaviour that could exist in real situations.

In total, 131 complete commands were tested. A compilation of the results obtained for each category is shown in Table 2. We listed the command's occurrence, evaluating if the voice command was received and interpreted by the ground station with a margin of error lower than five per cent relating to the voice recognition. If the result was outside this margin, the command was ignored and considered wrong.

Command	***Occurrence***	***Right***	***Wrong***
Left	27	25	2
Left front	25	20	5
Front	28	24	4
Left back	29	25	4
Back	22	19	3
TOTAL	131	113	18
Upper	23	23	0
Lower	22	22	0
Same	40	38	2
Below	27	24	3
Under	19	17	2
TOTAL	131	124	7
Far	42	41	1
Near	40	39	1
At	49	45	4
TOTAL	131	125	6

Table 2: Results of the tests

Analysing each one of the lines presented on Table 2 we can see that the *at* command had 4 wrong interpretations of 49 occurrences. Therefore the error rate of *at* command was greater than the other rates in the same category, such as *far* or *near*, which had just one wrong interpretation in each case. On the horizontal dimension, commands consisting of two words had the highest error rate, such as *left-front* and *left-back*, which had five and four wrong interpretations respectively of 25 and 29 occurrences. This was probably due to the existence of two other commands with the same words ending: (*front* and *back*), generating ambiguity. The analysis of the vertical dimension showed that the commands *upper* and *lower* had zero misinterpreted occurrences. That information was relevant when we take into account that the command *upper* and *lower* did not have other similar commands when looking at the phonetic point of view. It shows that the misinterpretation was probably due to noise present in voice recognition, not to the theory involved in the approach. These results showed that the error rate as less than five per cent on the vertical and radial dimensions. In the horizontal dimension, we obtained an error rate of more than ten per cent.

6 Conclusion

In this research, we bridged the gap between qualitative communication in an HRI setting using voice commands and the Qualitative Ego Sphere model as a basis of space qualitative information. The results showed the necessity of increasing the precision of our system, but also that our objective of simplifying the interaction between humans and robots has been achieved.

One of the contributions that can be related to this study is the accessibility improvement of non-specialist users to complex systems, like UAVs - *Unmanned Aerial Vehicles*. Using the approach presented in this paper, everyone able to pronounce the correct sequence of commands is capable of controlling the system successfully, and all the work with stabilisation will be the responsibility of the autonomous system itself. Another important contribution was facilitating the location requests to the quadrotor using quantitative information. In this case, for instance, the vehicle can be requested to go *near* or *far* the objective, using qualitative expressions, without the need of receiving the precise distance and coordinates of the goal location. This can be an advantage in emergency situations, where the answer time may be critical.

The lower error ratio obtained in the tests suggests the efficacy of the method investigated in this paper, but also brings atop the discussion about the equipment used on the system. With more precise instruments, such as using infrared sensors to filter the overall results, and some improvements on the code we can develop a system more consistent and achieve a higher level of communication between humans and a robot.

References

Markus Achtelik, Abraham Bachrach, Ruijie He, Samuel Prentice, and Nicholas Roy. 2009. Autonomous navigation and exploration of a quadrotor helicopter in gps-denied indoor environments. In *First Symposium on Indoor Flight*, 2009. Citeseer.

Kazumi Aoyama and Hideki Shimomura. 2005. Real world speech interaction with a humanoid robot on a layered robot behavior control architecture. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3814–3819. IEEE.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Anthony G. Cohn and Jochen Renz. 2008. Chapter 13 qualitative spatial representation and reasoning. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 551–596. Elsevier.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Christian Dondrup, Nicola Bellotto, Marc Hanheide, Kerstin Eder, and Ute Leonards. 2015. A computational model of human-robot spatial interactions based on a qualitative trajectory calculus. *Robotics*, 4(1):63–102.

Julien Fleureau, Quentin Galvane, Francois-Louis Tariolle, and Philippe Guillotel. 2016. Generic drone control platform for autonomous capture of cinema scenes. In *Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pages 35–40. ACM.

Antonio Franchi, Cristian Secchi, Markus Ryll, Heinrich H Bulthoff, and Paolo Robuffo Giordano. 2012. Shared control: Balancing autonomy and human assistance with a group of quadrotor uavs. *IEEE Robotics & Automation Magazine*, 19(3):57–68.

Bruce Johnson. 2012. *Professional visual studio 2012*. John Wiley & Sons.

John D. Kelleher and Simon Dobnik. 2022. Distributional semantics for situated spatial language? functional, geometric and perceptual perspectives. In J.-P. Bernardy, R. Blanck, S. Chatzikyriakidis, S. Lappin, and A. Maskharashvili, editors, *Probabilistic approaches to linguistic theory, CSLI Publications*, page 319356. Center for the Study of Language and Information, Stanford university, Stanford, California, USA.

María Malfaz and Miguel A Salichs. 2004. A new architecture for autonomous robots based on emotions. In *Fifth IFAC Symposium on Intelligent Autonomous Vehicles*.

Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer, and Marc Pollefeys. 2011. Pixhawk: A system for autonomous flight using onboard computer vision. In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 2992–2997. IEEE.

Reinhard Moratz and Marco Ragni. 2008. Qualitative spatial reasoning about relative point position. *Journal of Visual Languages & Computing*, 19(1):75–98.

D. H. Perico, P. E. Santos, and R. A. C. Bianchi. 2021. Guided navigation from multiple viewpoints using qualitative spatial reasoning. *Spatial Cognition & Computation*, 21(2):143–172.

Alexander R Perry. 2004. The flightgear flight simulator. In *Proceedings of the USENIX Annual Technical Conference*.

Jochen Renz, Debasis Mitra, et al. 2004. Qualitative direction calculi with arbitrary granularity. In *PRICAI*, volume 3157, pages 65–74.

Adam Richard-Bollans, Luca Gmez lvarez, and Anthony G. Cohn. 2020. Modelling the Polysemy of Spatial Prepositions in Referring Expressions. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*, pages 703–712.

Edilson J Rodrigues, Paulo E Santos, Marcos Lopes, Brandon Bennett, and Paul E Oppenheimer. 2020. Standpoint semantics for polysemy in spatial prepositions. *Journal of Logic and Computation*, 30(2):635–661.

Felipe Martino Esposito Rodrigues, Paulo E Santos, and Marcos Lopes. 2016. Communication of spatial expressions on multi-agent systems using the qualitative ego-sphere. In *Control and Automation (ICCA), 2016 12th IEEE International Conference on*, pages 25–30. IEEE.

Hazim Shakhatreh, Ahmad H. Sawalmeh, Ala Al-Fuqaha, Zuochao Dou, Eyad Almaita, Issa Khalil, Noor Shamsiah Othman, Abdallah Khreishah, and Mohsen Guizani. 2019. Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges. *IEEE Access*, 7:48572–48634.

Kaili Sun, Xudong Luo, and Michael Y. Luo. 2022. A survey of pretrained language models. In *Knowledge Science, Engineering and Management: 15th International Conference, KSEM 2022, Singapore, August 68, 2022, Proceedings, Part II*, page 442456, Berlin, Heidelberg. Springer-Verlag.

Nico Van de Weghe, Bart Kuijpers, Peter Bogaert, and Philippe De Maeyer. 2005. A qualitative trajectory calculus and the composition of its relations. In *GeoSpatial Semantics*, pages 60–76, Berlin, Heidelberg. Springer Berlin Heidelberg.