Towards 3D Simulation to Validate Autonomous Systems Intervention in Disaster Management Environment

Tullio Tanzi[1][0000-0001-5534-7712] and Matteo Bertolino[1][0000-0003-1043-9036]

[1] LTCI, Télécom Paris
Institut Polytechnique de Paris
Paris – France
firstname.lastname@telecom-paris.fr

Abstract. The use of autonomous robots either on the ground (i.e., Rover) or flying (i.e., Drone) constitutes a major progress in the support of a crisis. To work properly and to reach the desired level of autonomy, they have to be correctly configured though. Indeed, errors on robot configuration can lead to imprecise or erroneous data and, consequently, erroneous decisions can result from them. Before the beginning of the mission, it is important also to achieve a strong level of confidence about the usage of the sensors (for example, LIDARs) with respect to the context of the mission. Many aspects of these validations cannot be performed during the mission, for example verifying the behaviour of a rover following a strong collision with an external actor (such as debris) that can potentially damage or break some components. Moreover, during a real mission it is not always possible making huge modifications in the system configuration. In this respect, simulating the behaviour of the system in a virtual environment, similar to the real physical world, can constitute a good validation approach before the mission. These simulations allow to validate the behaviour and the configuration of the system as well as the most appropriate equipment of it.

Keywords: Autonomous System, Simulation, Models, Disaster.

1 Introduction

With the growing improving and reliability of technologies, we can rely on information technology (IT) to elaborate and refine a disaster recovery management strategy. A disaster recovery management strategy is intended to be the set of plans, processes and techniques to be implemented, with the final goal to saving lives as well as finding survivors or let the life restarts back to normal in the shortest possible time [1]. After a disaster, the impacted area could have suffered huge mutations due to terrain's alterations or presence of debris, for example. Some information can be obtained only after an in-place reconnaissance and not easily. The communication with the local inhabitants may be hindered by physical constraints or by the fact that people abandoned the area after the catastrophe. Even a local inspection presents strong limitations. For example, the poor visibility range can result in erroneous decisions based on incomplete or erroneous data. However, completeness and correctness information are necessary for the decision-making process that precedes the emergency intervention [2].

1.1 Benefits from robot usage

In this respect, the usage of Autonomous Systems (A.S.) such as rovers or drones can assist the recovering operations for many tasks. For example, A.S. can be used for Rapid-Mapping or to scan the affected area for finding survivors, among others. Figure 1 contains the 3D CAD view of our ArcTurius Rover [3], an A.S. for post-catastrophe humanitarian mission. It has been designed and developed by LabSoc, a research group on complex digital electronic systems from LTCI laboratory of Télécom Paris. ArcTurius Rover has to work for several days in total autonomy, underground (subsoil and basement given by building ruins), searching for the presence of survivors. Its design implies many challenges. Specifically, the environment of the mission prevents the rover by communicating with the operational centre by using radio-navigation means. Moreover, the length of the mission in terms of time introduces the problem of energy consumption. Thus, the design and the configuration have to be strongly considered in order to achieve the desired behaviour while considering the power management. In the context of this article, ArcTurius is our golden standard in the study of the rover behaviour within a virtual environment. Towards 3D-based simulations, our goal is to validate the design and the configuration (specific to a particular mission) of ArcTurius in order to enhance the effectiveness of A.S. usage after a catastrophe. Our laboratory works in the design of techniques that facilitate the recovery process in an environmentally critical context. In this regard, we propose a new approach, based on 3D simulations of the real world that speeds-up the definition of a recovery management strategy. Our research work aims to get a role immediately after a disaster happens, while the rescue teams are approaching the place of the crisis. It targets two main objectives:

1. Finding the best design and the best configuration of the A.S. for the target mission. To work properly and to reach the desired level of autonomy, A.S. have to be correctly designed and configured. Indeed, errors on robot configuration can lead to imprecise or erroneous data and, consequently, erroneous decisions can result from them. Before the beginning of the mission, it is important to achieve a strong level of confidence about the usage of the sensors [3, 4, 5] (for example, LIDARs) with respect to the surrounding world. For instance, we can evaluate the positioning of a LIDAR in order to minimise the impact of external noise, or whether the terrain's discontinuity perturbs data acquisition [6].
2. Acquiring a better knowledge of an area that the rescue teams do not know or do not know more because of environmental alterations. While the recovery teams are reaching the crisis area, A.S. can autonomously operate per-

forming the above mentioned tasks. Their result will help the definition of a strategy, even complex, or they can help modifying run-time an existing plan [2] [7].

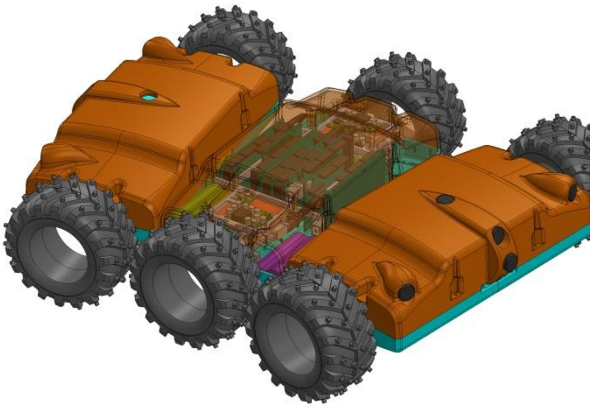

Fig. 1**.** ArcTurius Rover: autonomous system for post-catastrophe humanitarian mission (3D CAD view).

The configuration of an A.S. depends by some key points that include: (i) the configuration of the devastated area and ii) the main objectives of the SaR (*Search and Rescue*) mission such as damage assessment, people search and location, etc. The choice of the right equipment such as the set of sensors to be mounted as well as the best physical placement is a non-trivial task. A bad choice in this step may make the A.S. not reliable as expected.

2 Towards a new approach

We propose a 3D-simulation, which the direction of the rescue team can perform before their arrival. Briefly, an initial map of the real world is taken, for example through satellites' data, then it is injected onto a graphical engine. We define through a model-driven engineering approach the physics of the terrain and the physics of the actors that populate the world. Environmental conditions are taken into account too. We need to model the design and the geometry of the A.S. under examination and to provide a description of its behaviour. In this respect, the modelling of sensors and actuators, part of the A.S. and that interact with the external world, plays a main role. Through a physical engine, we are allowed to rapidly testing algorithms, designing robots and simulating their behaviour in realistic scenarios. The 3D vision enhances and speeds-up the comprehension of the designers.

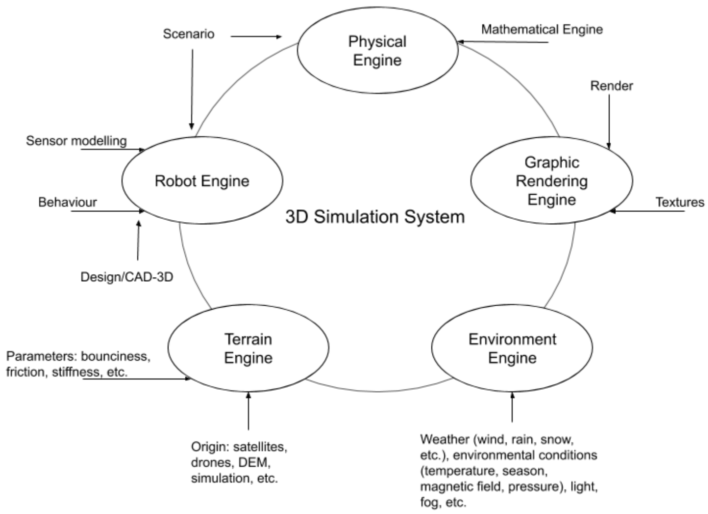

Fig. 2. 3D Simulation System general architecture

Figure 2 shows an overview of the proposed simulation system. The main components are:

1. Graphical engine that includes the rendering of textures, lights, shadows, etc.
2. Physical engine that allows realistic physics simulation and that is able to interact with the graphical engine with computer animation API
3. A comprehensive description (architecture, physics and behaviour) of the A.S. under examination. Sensors and actuators are included in this component of the system
4. An engine able to generate realistic terrain data
5. An engine that created realistic environment actors and conditions close to the real conditions due to a catastrophe.

Moreover, we need a way to establish the communication between all the components of the system. Even though we take into account the communication as part of the system in this article, we chose to do not list it as an explicit actor in Figure 2.

2.1 Robot modelling

In order to correctly reproduce the behaviour and the interaction of an A.S. in a surrounding world, we have to provide to the simulation system the following inputs: (i) architecture, (ii) physical properties, (iii) behaviour and (iv) sensors and actuators. The architecture of an A.S. is intended to be the shape and the geometry of each component that is part of the A.S. In this respect, the CAD models in Figure 1 can be a possibility. Providing the architecture of each mechanical component is not enough though. The domain of collision of each part of the system has to be defined as well as the physical properties of it. With physical properties we intend, for example, the characteristics features of rigid bodies: the respect of kinematics laws, friction, coefficients that describe an impact, etc. In this respect, modelling the junctions between components enhance the reliability of the model, because they play a key-role in the physical integrity of the system after a collision. A possible goal of the system is checking and analysing the behaviour of the A.S. during its interaction with the surrounding world. The system accepts as an input a program that has the role to move the A.S. during the simulation. In the context of rover ArcTurius, such program moves the representation of the rover within a virtual environment applying a force to the junctions that connect the wheels with the chassis of the rover, basing its decisions on the data acquired during the simulation. Even though sensors are logically part of an A.S., they have a different nature. For this reasons, we treat them in section 2.2.

2.2 Sensors and noise

Sensor output can be captured during the simulation and analysed through third-part programs. Even for sensors an architectural description has to be provided. However, because of they interact with the external environment by acquiring data, the description of physical properties of their design is negligible with respect to A.S. modelling. Thus, the behaviour of sensors plays a key-role for a realistic representation of the system. Except for few features (such as the acquiring frequency), each sensor has quite unique features that have to be taken into account separately. With respect to sensor modelling, some parameters to be taken into account while modelling a laser sensor include:

i) Physical shape,
ii) Relative poses with respect to A.S. components ,
iii) Number of samples per unit of time,
iv) Angular resolution ,
v) Minimum and maximum distance ,
vi) Noise.

Sensors are noisy components and we have to consider the noise to enhance the realism of the simulation. It is possible to model the inferences who affect them through a Gaussian distribution with a moment parameterization (i.e., providing mean and covariance of the distribution) [8]. Data coming from sensors can be analysed through third-part tools or used to take decisions. In this respect, Figure 4 shows the capturing of laser data through Rviz [9].

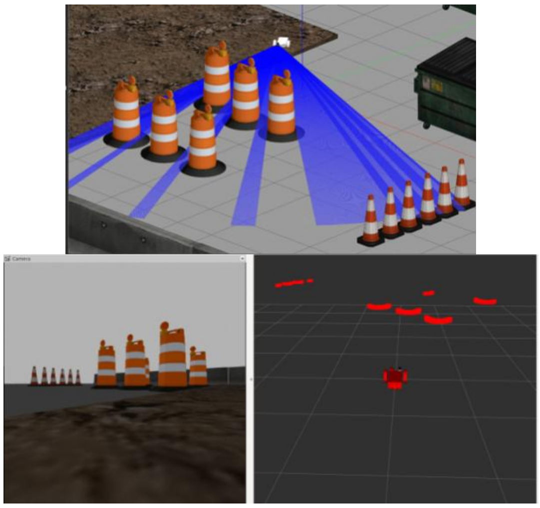

Fig.3. Example of Sensor (LIDAR) model.
Top global view. Bottom, left: video view, right: 3D scene reconstruction

2.3 Terrain generation and modelling

In SaR missions ground characteristics take a crucial role. Indeed, a correct representation of the ground permits to answer to many questions about the configuration of the A.S. during the simulation phase. Our simulation system is intended to speed-up the configuration of A.S. that support rescue teams during a mission. In this respect, rescue teams are mainly composed by rescuers and fire-fighters with limited engineering competences. The goal of this part is to avoid huge modification in A.S. configuration and mechanics once their arrival in catastrophe place. First of all, this would constitute a waste of time. Moreover, the environmental conditions do not always allow an easy intervention. A proper modelling of the terrain permits to preliminarily evaluate some issues related to the interaction of the A.S. with the surrounding world. Taking as a reference ArcTurius rover, we can evaluate if the depression of the terrain involve in a roll-over of it, or whether wheels of a different size are more appropriated for the mission, for example. Performing these kinds of simulations in advance permits rescuers to save time and to avoid a task for which they have limited competences. In our simulation system, realistic terrains can be provided as a input by processing satellites or airborne images, for example. In this respect, Digital Elevation Models (DEM) is a data format generated by combining sources from different sources, such as LIDAR, radar, cameras, photogrammetry, InSAR, land surveying, etc.) [10]. DEMs show surface elevation data, sampled at regularly-spaced horizontal intervals. If the terrain elevation is represented as a grid of elevations (raster), DEM can be seen as a grey-scale height-map. In height-maps, elevation data are represented by associated the colour of a grey-scale pixel with an elevation. Specifically, a white pixel corresponds to the point of maximum elevation, whereas a black pixel represents a point characterized by the minimum depression for the considered ground.

Figure 4 propose a way to constitute a 3D ground environment model that we used for the simulation for rover ArcTurius. The terrain represents an area in the south-west of Haiti and it has been generated by merging data coming from ASTER [11], USGS NED [12] and SRTM30+ [13]. ASTER is a well-known survey of elevation of Earth, characterized by a high coverage and resolution (about 30 meters). SRTM30+ provides coarse data (900 meters of resolution) and they are used to the general contours of the land. USGS NED provides precise elevation data (up to 10 meters of resolution) for zones close to United States geography.

Terrains generated from DEMs files do not usually contain external actors that can be useful for simulation purposes, such as rocks, rubble, fails, etc. However, they can be integrated in the ground. A possibility is to use a Drone that runs a low altitude to map them in a precise way. However, they can be generated and integrated in a realistic way. We can use mathematical functions, common in remote sensing applications, in order to estimate the roughness of the terrain. After, the physical properties of obstacles (such as density, bounciness, etc.) have to be defined, at least providing a range of values for each parameter. Finally, Monte Carlo method can be used to generate random distribution of these actors distributed over the terrain. To physically see the obstacles populating the world, a 3D description of them has to be provided too. Figure 5 schematically resumes the steps for the generation of terrains based on DEM models.

Fig.4. Ground model used in ArcTurius project

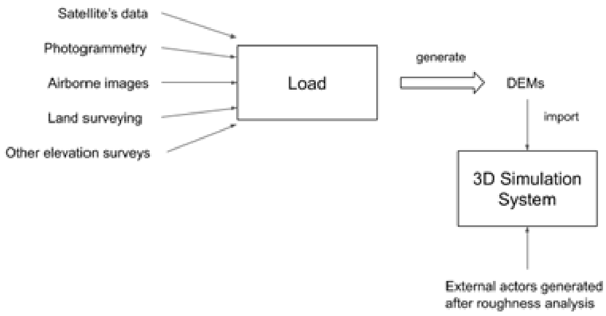

Fig.5. Ground model provide chain (DEM based example).

2.4 Environment parameterization

Environmental conditions can dramatically change the results given by a simulation and they have to be modelled in the more realistic way possible. For example, a simulation that involves the usage of A.S. equipped with LIDARs can suffer from the light in particular times of the day. By properly modelling the light, it is possible to derive a different position of LIDARs with respect to the chassis of the rover depending by the time of the day. For example, lights can be described specifying position, orientation, attenuation factors and direction, among others. Many other perturbations can derive by the different state of the environment, and each of them would require a strong analysis performed by a physician. Some of them include weather conditions, such as rain, snow or wind, temperature, magnetic field, other forces acting in the system, temperatures, etc. Through simulation, it is possible to evaluate the behaviour of an A.S. in a post-disaster area also considering external environment conditions that can limit the domain of validity of the simulation. A huge realism is obtained at cost of a precise physical modelling, which is usually done by people expert in the domain.

3 Graphical engine, Physical engine and Communications

Many graphical engines are available in the domain. For several reasons, only hinted in this paper, we chose Gazebo-3D [14]. Gazebo-3D is a graphic engine that includes a physical engine. It is able to perform 3D physical simulations while displacing a rover within a surrounding virtual world. Moreover, it can be used to rapidly evaluate different kinds of algorithm, to design robots and to simulate their behaviour in realistic scenarios. For that, it is necessary to realise realistic models of rover, sensors, actuators, environment conditions and external actors. Models are expressed with a XML-like syntax following SDF specifications [15]. They have to describe the physical properties (e.g., shape, collision domain, kinematics, friction, etc.) of the different actors that compose the simulation. It is possible to address a behaviour through the usage of plugins. For example, sensors shall transmit their measurements to a third-part analyser. Moreover, rover shall be able to move in a external environment and to interact with it, eventually basing its decisions on the values taken by the sensors. Gazebo-3D was chosen because of the experience of our laboratory on it and because it merges the graphical and the physical engine while maintaining a strong level of customization. Indeed, starting from the capabilities offered by Gazebo-3D, it is possible to define custom components able to interact with the simulator. The customization can occur both at graphical and behavioural level. Even though its expressiveness is very powerful, it is not useful whether not supported by a strong modelling of system features as well as an appropriate behaviour description of the various components through the usage of plugins. The latter can use Gazebo API to interact with the engines and can be developed in a custom way to describe the behaviour of rovers/drones, sensors, external actors, etc.

To assume the communication between the different parts of the simulation system we chose Robot Operating System (ROS) [16]. ROS is commonly used by the scientific community and despite its name, rather than a classical Operating System, it is more a framework that contains libraries, code, modules, configuration files, third-part software in support to robotics development. Its main goal it is to provide a standard communication interface to the different, heterogeneous elements that compose a robotics system. The latter are seen as “Nodes”. Nodes can produce data, consume data or both. The exchange of data happens following a standard syntax, even though nodes can have a very different nature each other. In particular, the exchange of data may happen in a continuous way (e.g., sensors nodes that send their value to a 3D-visualizer node) or they can occur one-off (e.g., a master node that express a request and wait for a reply to a slave node). In the first case, the nodes exchange data through ROS messages sent over ROS topics, with a reference paradigm similar to the Publisher/Subscriber. In the second case ROS messages are sent over artefacts named ROS services, with a reference model Client/Server. In our respect, ROS is used to connect all the elements intrinsic to the simulation (such as rover, environmental actors, sensors, etc.) and the data analysis tools that we use for the evaluation of the measurements.

4 Illustration of the Concept

The vast domain of this approach allows the representation of several and different case studies. A possible usage of this 3D simulation system permits to evaluate the positioning of a sensor, for example a camera, with respect to the chassis of an A.S. (such as a Rover). The purpose of the simulation is to find the best position for the camera, i.e. the position that minimises the number of useless samples taken during the mission. The first part of the simulation is ran in a flat floor, which does not hinder the displacement of the rover. The model of the rover is provided as input to Gazebo-3D as well as the description of its components. Initially, the camera is modelled as a solid figure (such as a cube) placed in a small turret located at the top of the rover chassis. The behaviour associated to the rover is a program that moves it following some criteria unspecified in this article, until finding a target. During the movement, the rover acquires pictures through a camera, which are sent through ROS to a node (that can be also physically located within the rover) that analyses them through the instruments given by OpenCV libraries [17]. Then, we repeat the simulation in a terrain characterised by a high index of roughness (e.g., there is plenty of debris and rocks). Depending by features such as the size and the properties of the wheels or the centre of mass of the rover, the presence of debris or rocks can modify the linear path of the A.S. This can involve the rover to acquire blurry pictures, crooked pictures or pictures that are useless with respect to the target recognizing algorithm. Consequently, it erroneously risks to miss the target. With respect to this illustration, the engineers responsible for the set-up of the A.S. can take countermeasures before the mission exploiting the results of the 3D simulation. For example, they can perform considerations about the positioning of the camera in the A.S. for specific terrains. Another possibility can be the replacement of the wheels of the rover with other that guarantee a better stability in such kind of terrains . This can enhance the utility of the A.S. and reduce the errors of the A.S during a mission.

5 Discussion and Conclusion

The use of autonomous engines either on the ground (Rover) or flying (Drone) is without any doubt an improvement. They are able to attain unattainable and dangerous locations. Moreover, they are less sensitive to both environmental conditions such as meteorology and stressful situations for human beings. However, in order to achieve this autonomy several difficulties have to be resolved. In the first hours after a disaster, during the rescue team deployment, our 3D-based simulation system allows a real gain in terms of timing. In conclusion, from the proposed approach we can expect to:

- Increase the knowledge of the devastated area
- Validate the A.S. behaviour (moving, data acquisition, etc.)
- Validate the adaptation of the payload sensors configuration for this (these) mission(s)

This is a first work towards the proposal of a new approach than can support the support rescue missions that expect a contribution of A.S. Today we are able to generate terrain data based on satellites and DEM model, to customize them and to import them in the simulation environment. Moreover, we can correctly model rigid bodies and their physical properties, such as the mechanical parts that compose a rover with their interconnection points (junctions). We are able to associate a behaviour to the rover, to model the most common sensors (such as camera, LIDARs, odometers) and display their results. On the other hand, more work has to be done in the context of environment conditions. Part of the future work will include the reaction of the terrain (intended as the modification of its physical parameters) to a natural event such as rain on snow. This work will open the door to many others scenarios in the context of simulation in support to disaster risk reduction [18, 19].

References

[1] Biswajit, Mukhopadhay & Bhattacherjee, Buddhadev. (2015). Use of Information Technology in Emergency and Disaster Management. American Journal for Environmental Protection. 4. 101-104. 10.11648/j.ajep.20150402.15.
[2] Tanzi, Tullio & Isnard, Jean. (2019). Autonomous system for data collection: Location and mapping issues in post-disaster environment. Comptes Rendus Physique. 10.1016/j.crhy.2019.03.001.
[3] Tanzi, Tullio & Isnard, Jean. (2019). Autonomous system for data collection: Location and mapping issues in post-disaster environment. Comptes Rendus Physique. 10.1016/j.crhy.2019.03.001.
[4] Madhu Chandra and Tullio Joseph Tanzi, (2017). "Drone-Borne GPR Design: Propagation Issues," in Journées scientifiques de l'URSI-France (JS'17)
[5] Sylvie Servigne, Yann Gripay, Ozgun Pinarer, John Samuel, Atay Ozgovde & Jacques Jay, (2016). "Heterogeneous Sensor Data Exploration and Sustainable Declarative Monitoring Architecture: Application to Smart Building". First International Conference on Smart Data and Smart Cities, 30th UDMS, 9 septembre 2016, Split (Croatie), pp. 97-104, doi : 10.5194/isprs-annals-IV-4-W1-97-2016.
[6] Tanzi, Tullio Joseph; Roudier, Yves; Apvrille, Ludovic. (2015). "Towards a new architecture for autonomous data collection". ISPRS Geospatial Week 2015: Workshop on civil Unmanned Aerial Vehicles for geospatial data acquisition. 1-2 October, 2015. La Grande Motte (Montpellier), France.
[7] Stormont, Daniel & Allan, Vicki. (2009). Managing Risk in Disaster Scenarios with Autonomous Robots. Journal of Systemics, Cybernetics and Informatics. 7.
[8] Sebastian and Burgard, Wolfram and Fox, Dieter Thrun, Probabilistic Robotics.: The MIT Press, 2005.
[9] Kam, HyeongRyeol & Lee, Sung-Ho & Park, Taejung & Kim, Chang-Hun. (2015). RViz: a toolkit for real domain data visualization. Telecommunication Systems. 60. 1-9. 10.1007/s11235-015-0034-5.
[10] Chandra Madhu & Tanzi Tullio, (2014), Wave Propagation and Radar System. Aspects for Designing a "Drone borne" GPR for Humanitarian Application, "IEEE Conference on Antenna Measurements & Applications (CAMA)", Antibes, France.
[11] ASTER - Global Digital Elevation Map https://asterweb.jpl.nasa.gov/gdem.asp
[12] USGS DEM - National Elevation Dataset (NED)
https://en.wikipedia.org/wiki/USGS_DEM
[13] SRTM30+ - Shuttle Radar Topology Mission https://www2.jpl.nasa.gov/srtm/
[14] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat.No.04CH37566), Sendai, 2004, pp. 2149-2154 vol.3. doi: 10.1109/IROS.2004.1389727
[15] SDF data format - http://sdformat.org/
[16] Quigley, Morgan & Conley, Ken & P Gerkey, Brian & Faust, Josh & Foote, Tully & Leibs, Jeremy & Wheeler, Rob & Y Ng, Andrew. (2009). ROS: an open-source Robot Operating System. ICRA Workshop on Open Source Software. 3.

[17] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, "A brief introduction to OpenCV," *2012 Proceedings of the 35th International Convention MIPRO*, Opatija, 2012, pp. 1725-1730.
[18] Ludovic Apvrille, Tullio Joseph Tanzi, Yves Roudier, and Jean-Luc Dugelay. (2017) "Drone "humanitaire : état de l'art et réflexions," Revue Française de Photogrammétrie et de Télédétection, pp. 63-71, N 213-04-26. 2017. ISSN 1768-9791
[19] Tullio Joseph Tanzi, Madhu Chandra, Jean Isnard, Daniel Camara, Olivier Sebastien, Fanilo Harivelo. (2016) "Towards rone-Borne Disaster Management: Future Application Scenarios ". 06/2016; III-8:181-189. DOI:10.5194/isprs-annals-III-8-181-2016