

SWARM INTELLIGENCE IN DISASTER RECOVERY

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Technology Degree in Information Technology
By

IMMANUAL STANLEY (37120031)
JITHIN ABHISHEK (37120036)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMIL NADU**

MARCH 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" Grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.

Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344

www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project is the bonafide work of IMMANUAL STANLEY (Reg no:37120031), JITHIN ABHISHEK(Reg. No. 37120036) who carried out the project entitled **“SWARM INTELLIGENCE IN DISASTER RECOVERY”** under my supervision from November 2020 to April 2021.

Internal Guide

Dr.L.MARY GLADENCE

Head of the Department

Dr. R.SUBHASHINI, M.E.,Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **IMMANUAL STANLEY (Reg no:37120031), JITHIN ABHISHEK (Reg. No. 37120036)** hereby declare that the Professional Training Report on “**SWARM INTELLIGENCE IN DISASTER RECOVERY**” done by me under the guidance of guide **DR.L.MARY GLADENCE**, Department of Information Technology at Sathyabama Institute of Science and technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.T.Sasikala M.E.,Ph.d, Dean**, School of Computing, **Dr. Subashini M.E.,Ph.D.**, Head of the Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.L. Mary Gladence**, for this valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

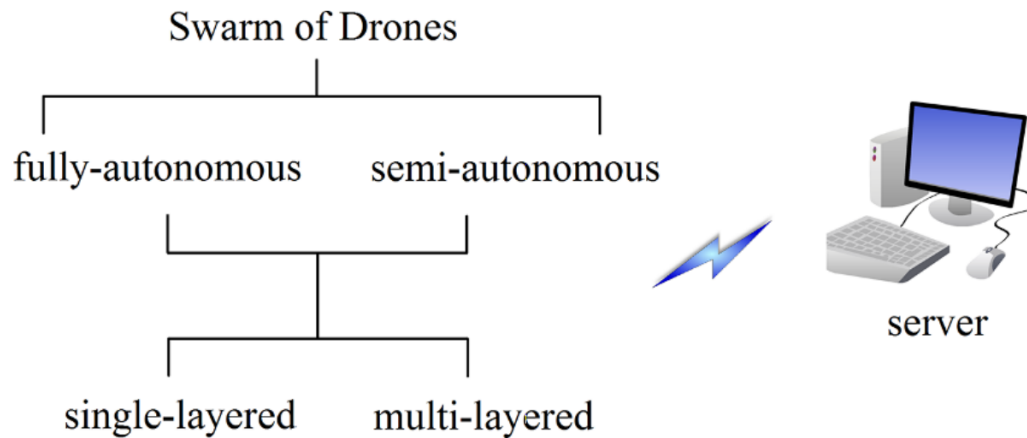
CHAPTER - 1

INTRODUCTION

1.1 Unmanned Aerial Vehicles (UAVs)

A swarm or fleet of Unmanned Aerial Vehicles (UAVs) is a set of aerial robots i.e., drones that work together to achieve a specific goal. Each drone in a swarm is propelled by a specific number of rotors and has the ability to vertically hover, take-off, and land (VTOL). The flight of the drones is controlled either manually, i.e. by remote control operations, or autonomously by using processors deployed on the drones. A common purpose for drones is a military one, but their civilian applications are attracting increased attention in the recent time. Indeed, low-cost drones and their swarms provide a promising platform for innovative research projects and future commercial applications that will help people in their work and everyday lives. Swarms of drones can be classified in different ways. illustrates fully and partly (semi) autonomous swarms. From another point of view, the classification can be envisioned in single layered swarms with every drone being its own leader and multilayered swarms with dedicated leader drones at every layer, which report to their leader drones at a higher layer; a ground-based server station is the highest layer in this hierarchy. In each swarm, every drone can have dedicated data collection and processing tasks with sufficient computing capability to execute these tasks in real-time. Its central processing takes place on the more performant server/base station or even in the cloud. The advances in the capabilities of sensors deployed on UAVs enable the use of drones for different new purposes, facilitating the creation of a new breed of applications and services in the sector of unmanned operations. UAVs have traditionally been employed in military surveillance missions. Versatile and low-cost drones have been utilized in aerial surveys, for monitoring and surveillance, in numerous fields such as geophysics and agriculture. For example, surveillance of a facility or environment might require updates of every movement detected after office hours. A large facility or environment would require a lot of manpower for thorough manual surveillance. Contrary to this, a swarm of drones can cover/monitor the region much more efficiently with a minimal manual

effort, by automatically and quickly sending an alert to the base station upon detection of movement.



Mechanical Design

For the controlled movement of the quadcopter in the presence of disturbance and uncertainties, the initial step is to construct a balanced drone with an adaptive computing platform. The targeted control system is designed by building a dynamic model of the drone, consisting of a set of mathematical equations dealing with all the acting forces on the system at a given time interval.

Classification of UAVs

UAVs, in the market, come with diversity in the number of propellers/rotors as illustrated in Table 1. Drones can also be grouped based on their size, range, and equipment. The sizes can be either nano, mini, regular, or large while the range can be either very close, close, short, mid, or endurance. Drones can be equipped with cameras, stabilizers, sensors, Global Positioning System (GPS), and/or First-Person View (FPV) accompanied by FPV goggles.

Categorization of UAVs based on the number of propellers.

Drones	No. of propellers
Tricopter	3
Quadcopter	4
Hexacopter	6
Octocopter	8

Drones are classified into four major types; fixed-wing, fixed-wing hybrid, single rotor, and multirotor as shown in Table 2. Fixed-wing drones are mainly used for aerial mapping and inspection of pipelines/ power lines. They are expensive and require skill training to operate. Though they require relatively more space for their launch and recovery, they are equipped to cover larger areas. This type of drones is not suitable for VTOL/hover and hence this makes them unfit for general aerial photography. However, they can stay in the air for up to 16 h by using gas engines as their power source.

Drones	Main features
Fixed-Wing	long endurance and fast flight speed
Fixed-Wing Hybrid	VTOL and long endurance flight
Single Rotor	VTOL, hover, and long endurance flight
Multirotor	VTOL, hover, and short endurance flight

In contrast, fixed-wing hybrid drones are a combination of manual gliding and automation. They are still in the development phase and not good at either hover or forward flight. They are commercially being used by Amazon for delivery purposes. On the other hand, single rotor drones have more mechanical complexity and operational risks such as vibration and large rotating blades. Therefore, they also require skill training of the operator. They are costly and have the competency of the heavier payload such as LiDAR sensor to be steered. For even longer endurance, they can be powered by a gas motor. Among the four types, the multirotor UAVs are the cheapest option available and easier to build. Such drones

are commonly used for basic purposes such as photography and video surveillance. They can be tricopters, quadcopters, hexacopters, or octocopters. However, these types of drones are not appropriate for larger-scale aerial mapping and longer distance monitoring due to their limited speed, flight time and energy efficiency. Currently available multirotor drones can fly up to 30 min with a normal lightweight payload such as a camera. The most commonly used UAV type (considering the number of propellers) is a quadcopter, and therefore our discussion in the following sections will focus specifically on this type.

1.2 Introduction of the project:

UAVs can save a lot of time and manpower by providing real-time aerial images of targeted locations. As a result, S&R teams can easily determine where exactly the help is most urgently needed. For example, different UAVs in a swarm can be equipped with different basic first aid kits that can be redirected and delivered to a person in need of medical assistance even before the medical team reaches him/her, resulting in a much better chance to save someone's life.

1.3 Objective

To Develop a Drone swarm of autonomous flying robots to cooperatively gather situational awareness data during the first few hours after a major natural disaster and to provide emergency communication network connection to people on the ground (in disaster areas, where there is no more connectivity), using swarms of flying robots

CHAPTER - 2

LITERATURE SURVEY

1.Title: “IEEE: 3D Optimal Surveillance Trajectory Planning for Multiple UAVs by Using Particle Swarm Optimization With Surveillance Area Priority ,” NRF , HU TENG, ISHTIAQ AHMAD, ALAMGIR MSM , AND KYUNGHI CHANG Year:2020

This Model evaluated the trajectories generated by the proposed trajectory planner, and introduced a multi-objective fitness function that has eight optimization indexes in terms of UAV maneuverability. The optimal trajectories are obtained by maximizing the fitness function values. Their simulation results prove that our proposals can obtain collision-free trajectories for multiple UAVs with high fitness values, and they show highly dynamic environmental adaptability..

2. Title : “Journal of Industrial Information Integration: Swarms of Unmanned Aerial Vehicles”, Anam Tahira , Jari Bölingb , Mohammad-Hashem Haghbayana , Hannu T. Toivonenb , Juha Plosila Year:2019

They presented a survey-type study on UAVs (drones) and swarms of UAVs, with a special focus on quadcopters. The mechanics, functionality, organization, modelling, applications, and autonomy aspects of such drones and drone swarms were discussed. Additionally, the paper included the result of an online survey in order to get a picture of public awareness regarding the use of drone technology..

3. Title : “IEEE: Energy-Efficient Formation Morphing for Collision Avoidance in a Swarm of Drones”, JAWAD NAVEED YASIN , SHERIF ABDELMONEM SAYED MOHAMED , MOHAMMAD-HASHEM HAGHBAYAN , JUKKA HEIKKONEN , HANNU TENHUNEN , MUHAMMAD MEHBOOB YASIN , AND JUHA PLOSILA Year:2020

They proposed a Thin-Plate Spline inspired formation maintenance algorithm for multiple UAVs, integrated with a collision avoidance capability. By the ability to

accelerate and decelerate on demand, the drones can efficiently reach their respective leaders or find their places in the formation when they start at random coordinates or wander off due to the presence of obstacles in their vicinity. The collision avoidance scheme is able to flexibly handle situations with multiple detected obstacles. They plan to extend the algorithm to handle the 3-dimensional movement along with the introduction of other environmental effects such as air drag on the individual nodes as well as on the overall shape of the swarm.

4. Title: “MDPI: Review of Unmanned Aerial Vehicle Swarm Communication Architectures and Routing Protocols”, Xi Chen, Jun Tang Year:2020

They introduced four communication architectures that can be deployed in UAV swarms, and also introduced three common intra-swarm communication architectures. They conclude that the multi-layer architecture combined with the meshed intra-swarm architecture is currently the most applicable communication architecture. Although the existing theoretical research and experiments have solved some technical problems in UAV swarm communication architectures and routing protocols, there are still many interesting, important and yet challenging open issues deserving further investigation. In addition, the UAV communication network should be compatible with other networks in specific mission scenarios. Therefore, the design of the new routing protocols should be able to effectively support the different transmission requirements in the diverse mission scenario

5. Title: “MDPI: Beehive-Inspired Information Gathering with a Swarm of Autonomous Drones” Alberto Viseras , Thomas Wiedemann , Christoph Manss, Valentina Karolj, Dmitriy Shutin and Juan Marchal Year: 2019

It presented a beehive-inspired multi-agent drone system for autonomous information collection to support the needs of first responders and emergency teams. The proposed system was designed to be simple, cost-efficient, yet robust and scalable at the same time. That included several unmanned aerial vehicles that can be tasked with data collection, and a single control station that acts as a data accumulation and visualization unit. The proposed design also proved experimentally to be robust against latter events. In particular, when a drone does

not return to the control station after a predefined timeout, its tasks can be reassigned either to spare drones or to other drones used in the system. This ensures that data will be continuously collected, even if only one drone is used in the system

CHAPTER - 3

AIM AND SCOPE OF THE PRESENT SYSTEM

EXISTING SYSTEM

UAVs have traditionally been employed in military surveillance missions. Versatile and low-cost drones have been utilized in aerial surveys, for monitoring and surveillance, in numerous fields such as geophysics and agriculture . For example, surveillance of a facility or environment might require updates of every movement detected after office hours. A large facility or environment would require a lot of manpower for thorough manual surveillance.

Why simulation?

- **Risk-free environment**

Simulation modeling provides a safe way to test and explore different “what-if” scenarios. The effect of changing staffing levels in a plant may be seen without putting production at risk. Make the right decision before making real-world changes.

- **Save money and time**

Virtual experiments with simulation models are less expensive and take less time than experiments with real assets. Marketing campaigns can be tested without alerting the competition or unnecessarily spending money

- **Visualization**

Simulation models can be animated in 2D/3D, allowing concepts and ideas to be more easily verified, communicated, and understood. Analysts and engineers gain trust in a model by seeing it in action and can clearly demonstrate findings to management.

- **Insight into dynamics**

Unlike spreadsheet- or solver-based analytics, simulation modeling allows the observation of system behavior over time, at any level of detail. For example, checking warehouse storage space utilization on any given date.

- **Increased accuracy**

A simulation model can capture many more details than an analytical model, providing increased accuracy and more precise forecasting. Mining companies can significantly cut costs by optimizing asset usage and knowing their future equipment needs.

- **Handle uncertainty**

Uncertainty in operation times and outcome can be easily represented in simulation models, allowing risk quantification, and for more robust solutions to be found. In logistics, a realistic picture can be produced using simulation, including unpredictable data, such as shipment lead times.

CHAPTER - 4

EXPERIMENTAL METHODS AND ALGORITHMS

HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list, especially in case of operating systems. The minimal hardware requirements are as follows,

1. PROCESSOR : intel core i5 8300h
2. RAM : 8 GB
3. PROCESSOR : 3.1 Ghz
4. PROCESSING SPEED : 600 MHZ
5. HARD DISK DRIVE : 1TB

SOFTWARE REQUIREMENTS

Software requirements deals with defining resource requirements and prerequisites that needs to be installed on a computer to provide functioning of an application. These requirements are need to be installed separately before the software is installed. The minimal software requirements are as follows,

1. FRONT END :Unity 3d
2. IDE : Visual Studio
3. SDK : OpenCV
4. Language: C#
5. OPERATING SYSTEM :WINDOWS 10

INTEGRATED DEVELOPMENT ENVIRONMENT

What is Unity3D?

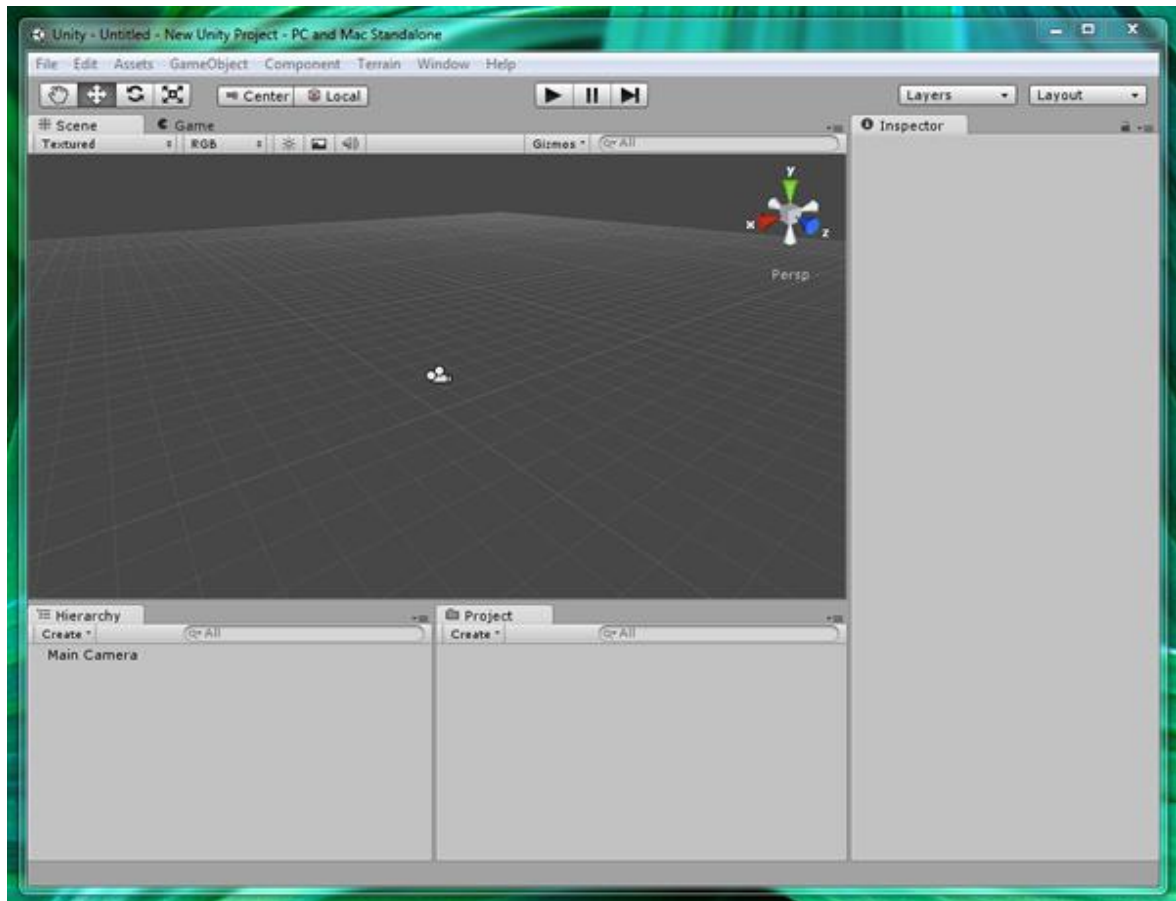
Unity3D is a powerful cross-platform 3D engine and a user friendly development environment. Easy enough for the beginner and powerful enough for the expert; Unity should interest anybody who wants to easily create 3D games and applications for mobile, desktop, the web, and consoles.

The Application

The Unity application is a complete 3D environment, suitable for laying out levels, creating menus, doing animation, writing scripts, and organizing projects. The user interface is well organized and the panels can be fully customized by dragging and dropping.

Scenes

Scenes are where you can drag in project assets and arrange them to make levels and game screens. The Hierarchy panel represents the contents of the current scene in a tree-like format. While the Scene panel is ideal for arranging your scene's assets in 3D space, the Hierarchy is where you'll spend most of your time actually organizing your scenes and keeping them tidy.



Assets

Assets are any resource your game uses. These include 3D models, materials, textures, audio, scripts, and fonts, to name a few. Other than a few simple objects such as cubes and spheres, Unity can't actually create most of these assets. Instead, they must be created externally using 3D modeling applications and painting tools and then imported into Unity.

Thankfully, Unity's asset importing is robust and intelligent. Traditionally, 3D game engines have usually been finicky things and are very particular about what files you give them, forcing developers to carefully convert all their files. Not Unity. It will accept all popular 3D file formats including Maya, 3D Studio Max, Blender and FilmBox with all the rigging, materials and textures intact. Unity also supports all common image file formats, including PNG, JPEG, TIFF and even layered PSD files directly from Photoshop. When it comes to audio, Unity supports WAV and AIF, ideal for sound effects, and MP3 and OGG for music.

Scripting

Scripts, known in Unity as behaviours, let you take assets in your scene and make them interactive. Multiple scripts can be attached to a single object, allowing for easy code reuse. Unity supports three different programming languages; UnityScript, C#, and Boo. UnityScript is similar to JavaScript and ActionScript, C# is similar to Java, and Boo is similar to Python. Depending on your background you may feel more comfortable with one or the other.

Debugging in Unity

Tweaking and debugging is efficient in Unity because all the gameplay variables are shown right as developers play, so things can be altered on the fly, without writing a single line of code. The game can be paused at anytime or you can step-through code one statement at a time.

Publishing

Unity is able to publish to Windows, OS X, and the web via the Unity Web Player. The Web Player is a browser plugin that works in all major browsers and offers the same performance available on the desktop

Benefits of data-oriented design with DOTS

The traditional GameObject-Component concept continues to work well because it's easy to understand for programmers and non-programmers alike, and easy to build intuitive UIs for. You add a Rigidbody Component to a GameObject and it will start falling, or a Light Component to a GameObject and it will emit light. And so on.

However, the Component system was written in an object-oriented framework and it creates challenges for developers when it comes to managing cache and memory in ever-evolving hardware.

Components and GameObjects are “heavy C++” objects. All GameObjects have a name. Their Components are C# wrappers on top of C++ components. This makes them easy to work with, however, it can come at a cost to performance because they potentially end up stored in an unstructured way. That C# object could be anywhere in memory. The C++ object can also be anywhere in memory. Things are not grouped together in contiguous memory. Every time anything is loaded in CPU for processing, everything has to be fetched from multiple locations. It can get slow and inefficient and therefore, require a lot of optimization workarounds.

Test, train, and validate in a real-time 3D virtual world

Unity Simulation is a managed cloud service that enables product developers, researchers, and engineers to easily and efficiently run thousands of instances of parameterized Unity projects in batch in the cloud.

By developing projects in Unity that represent real-world environments, parameterizing those projects for simulation, and running many simulated scenarios in the cloud faster than real time, you can accelerate your time to insight.

Key benefits

1. Easy, affordable testing and validation for AI algorithms

Test and validate the effectiveness of your AI algorithms in a virtual capacity, eliminating risks and reducing costs that come from real-world testing.

2. Fast, cost-effective training for machine learning

Generate pre-labelled synthetic data – at a fraction of the typical time and cost of constructing and capturing real-world data.

3. Better systems modeling

Efficiently simulate spatial environments across scenarios and over longer timescales than other types of simulation tools to holistically understand large, complex, and uncertain systems.

4. Zero hardware investment

Pay only for actual cloud usage, without monthly minimums or upfront investment. Because we manage the cloud service, there's no need to install and manage batch computing software or server clusters that you use to run your jobs, allowing you to focus on analyzing results and solving problems.

Cost

Indie developers rejoice, Unity is free! Well, almost. There is a Pro edition that comes with more features and tools, but will set you back \$1,500. Considering the feature set and how permissive the Unity publishing license is, this price is actually very reasonable. However, the free version will let you get your feet wet, build complete games, and even publish them to the desktop and the web without paying a cent! The only caveat is that games published under the free edition will have a small Unity watermark. Anything built in Unity will work exactly the same way in Unity Pro. This means you can choose to upgrade at any point if you need the additional features, or want to publish to more platforms such as iOS and Android.

SOFTWARE DEVELOPMENT KIT

Open CV

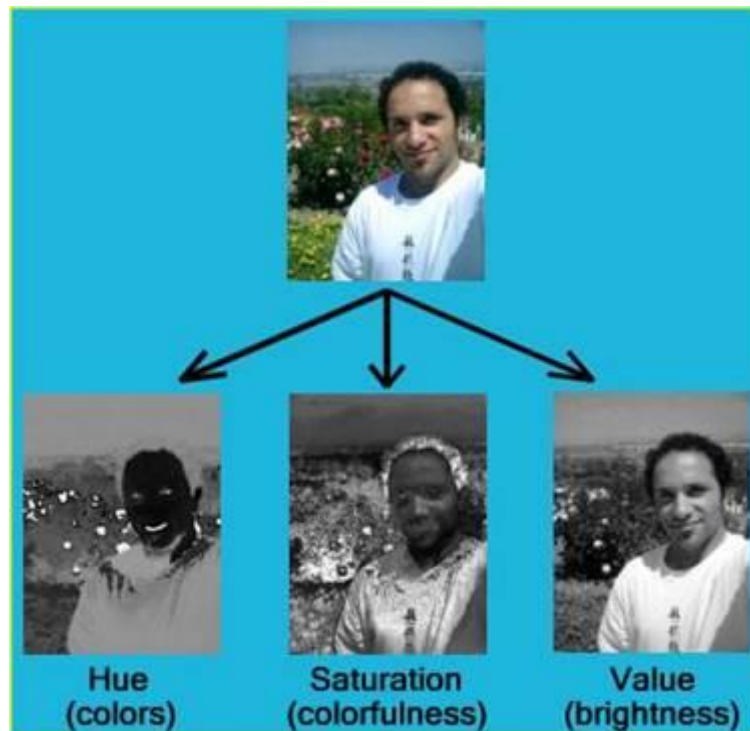
- **Computer Vision:** Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps

- **OpenCV :** OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as Numpuy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

Applications of OpenCV: There are lots of applications which are solved using OpenCV, some of them are listed below

- Face recognition
- Automated inspection and surveillance
- number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds
- Interactive art installations
- Anamoly (defect) detection in the manufacturing process (the odd defective products)
- Street view image stitching
- Video/image search and retrieval
- Robot and driver-less car navigation and control
- object recognition
- Medical image analysis
- Movies – 3D structure from motion
- TV Channels advertisement recognition

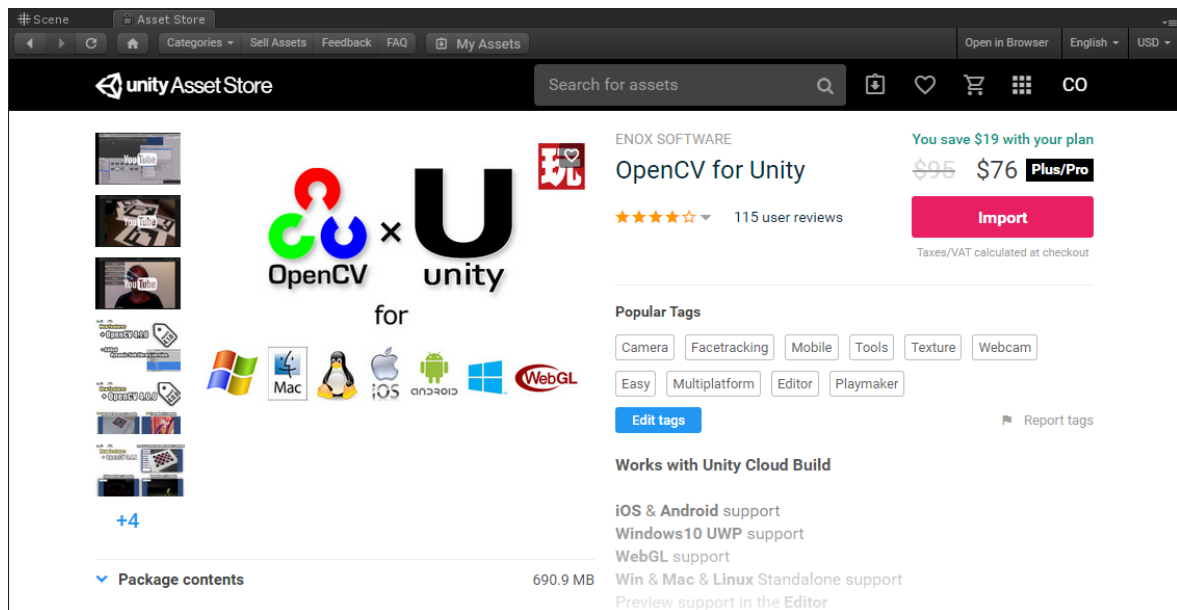


How Does A Computer Read An Image?

The computer reads any image as a range of values between 0 and 255. For any color image, there are 3 primary channels -red, green and blue.

OpenCV Functionality

- Image/video I/O, processing, display (core, imgproc, highgui)
- Object/feature detection (objdetect, features2d, nonfree)
- Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab)
- Computational photography (photo, video, superres)
- Machine learning & clustering (ml, flann)
- CUDA acceleration (gpu)



Import OpenCV

The fastest way to use OpenCV within Unity is to use Enox Software's OpenCV for Unity package on the Asset Store. You can use the free trial version for the tutorial, but you won't be able to build a standalone app.

How Unity Simulation works

1. Create your virtual environment

Create a Unity project that represents a real-world environment using Unity Pro, our real-time 3D development platform, or work with our professional services team for project creation.

2. Configure your Unity project for simulation

Parameterize your Unity project in ways that will change from run to run (e.g., positions or colors of assets, scene layout configurations) and specify simulation output data necessary for your end application.

3. Run simulations in the cloud

Define your simulation job, get a cost estimate, and execute it on the cloud.

4. Download output data

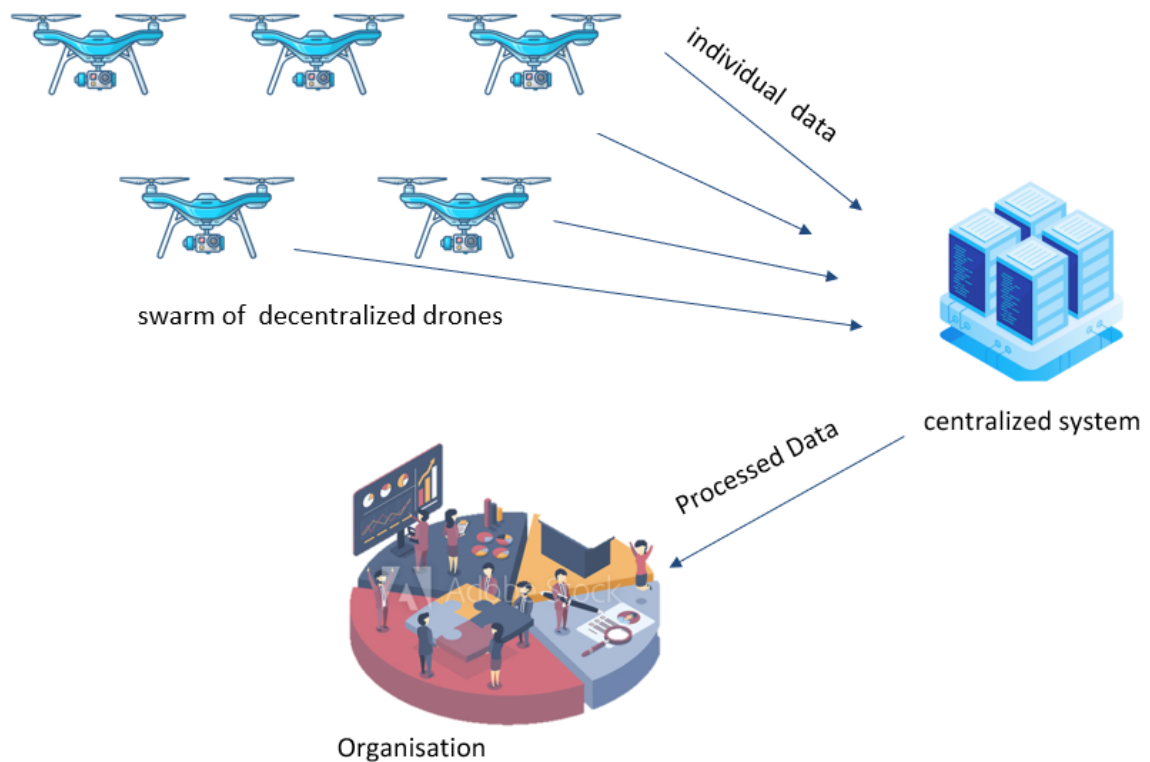
Download the data generated from the simulation – whether it's segmented images for machine learning training or logged data for AI testing or modeled systems – to use in your end application.

Automotive simulations using LG and Unity

- An autonomous vehicle must log at least 11 billion miles before it is considered road safe. It would take hundreds of years to reach that goal with real-world driving or running simulations locally.
- The LGSVL Simulator takes advantage of several advanced Unity features to simulate photorealistic digital twins of real-world environments with accurate vehicle and sensor models.
- With Unity Simulation and the LGSVL Simulator, developers can now train their machine learning models through all scenarios and edge cases in a matter of hours.

SYSTEM DESIGN

SYSTEM ARCHITECTURE



UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

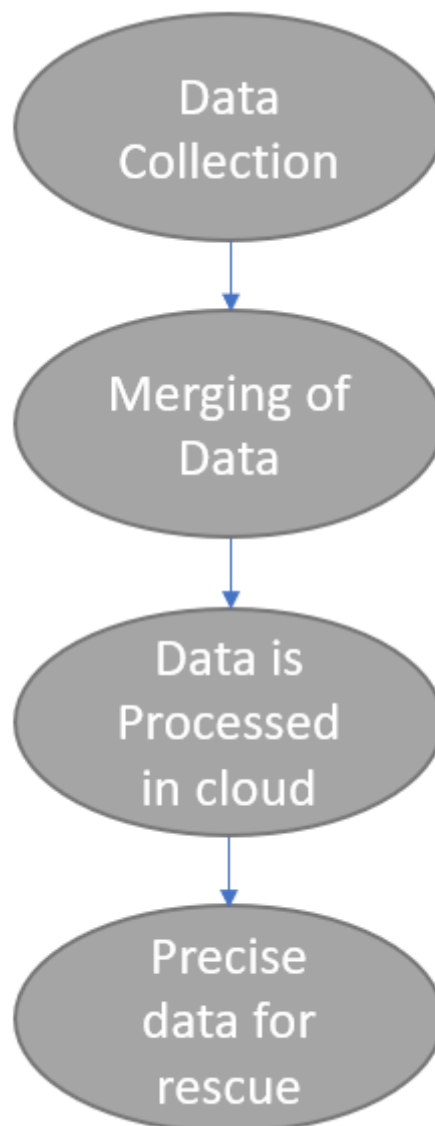
GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



CHAPTER - 5

RESULTS AND PERFORMANCE ANALYSIS

Data collection:

- Decentralized Drones are sent to fetch information about the affected area to detect the presence of life via cameras
- The Geolocation of the Area is fed to the server, And the according to the number of drones present, the path is determined by the user.
- By the user defined path, the drone will scavenge for people who are stuck in the places of flooded areas
- Using the quad-camara, or 360 camera , and open cv the drone will identify the people and tag them with their locations and store them in its physical memory

Transfer of Data to servers

- Each Drone holds a separate individual data, and the data is sent to server, which merges each part of data which the drones.
- The drone(s) will send the real time information and refresh the information on the cloud for a specified frequency of time
- If there are is more than one drone, the drones will very frequently refresh as it should not count the same people twice, in nearby areas where another drone is set up

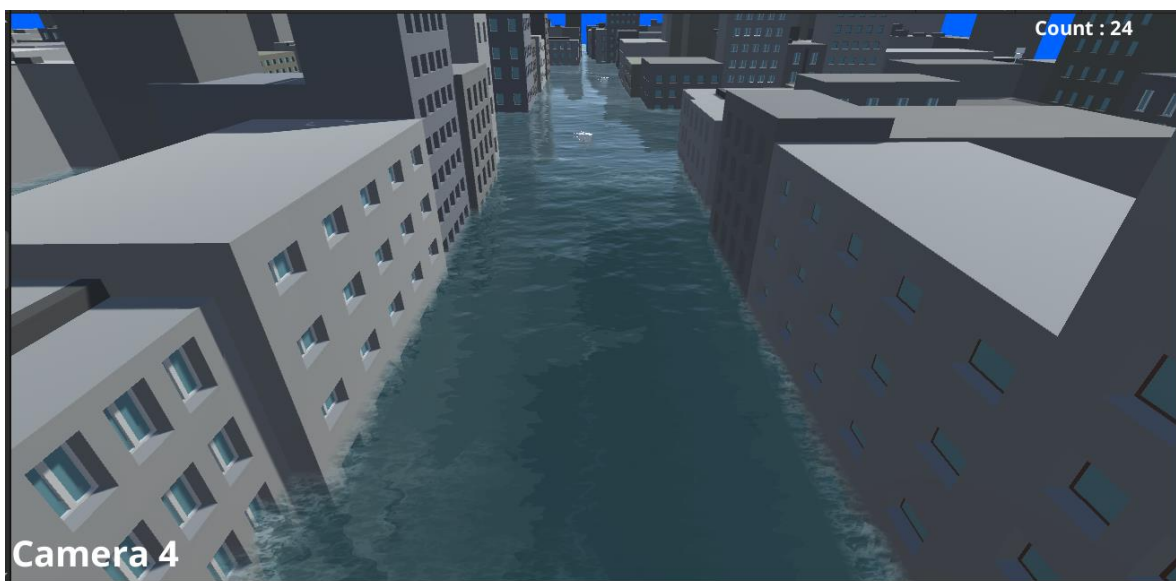
Data Processing:

- On the servers, the data is fed and using swarm intelligence, the data are plotted on a graph.
- On the server, or the cloud, the data that is being collected by the drone(s) are transferred in real time
- Information include the number of people, and with the areas where those people are found.

Collected Data:

- The graph that is made can be accessed by the organisation and can start with the required measures for the safety of people.
- Using the data, the rescue team can correlate and the other information like area of disaster, Time of flooding in the area, Intensity of flood, Total area and distance affected, Route to reach the affected area, Medium of communication, in the affected area, Number of rescue team(s) required and through these factors the team can help the affected people

Simulated Work



Total people on the scanned by the drone in the city : 24

TESTING

Software Testing

General

In a generalized way, we can say that the system testing is a type of testing in which the main aim is to make sure that system performs efficiently and seamlessly. The process of testing is applied to a program with the main aim to discover an unprecedented error, an error which otherwise could have damaged the future of the software. Test cases which brings up a high possibility of discovering and error is considered successful. This successful test helps to answer the still unknown errors.

TEST CASE

Testing, as already explained earlier, is the process of discovering all possible weak-points in the finalized software product. Testing helps to counter the working of sub-assemblies, components, assembly and the complete result. The software is taken through different exercises with the main aim of making sure that software meets the business requirement and user-expectations and doesn't fails abruptly. Several types of tests are used today. Each test type addresses a specific testing requirement.

Testing Techniques

A test plan is a document which describes approach, its scope, its resources and the schedule of aimed testing exercises. It helps to identify almost other test item, the features which are to be tested, its tasks, how will everyone do each task, how much the tester is independent, the environment in which the test is taking place, its technique of design plus the both the end criteria which is used, also rational of choice of theirs, and whatever kind of risk which requires emergency planning. It can be also referred to as the record of the process of test planning. Test plans are usually prepared with signification input from test engineers.

(I) UNIT TESTING

In unit testing, the design of the test cases is involved that helps in the validation of the internal program logic. The validation of all the decision branches and internal code takes place. After the individual unit is completed it takes place. Plus it is taken

into account after the individual unit is completed before integration. The unit test thus performs the basic level test at its component stage and test the particular business process, system configurations etc. The unit test ensures that the particular unique path of the process gets performed precisely to the documented specifications and contains clearly defined inputs with the results which are expected.

(II) INTEGRATION TESTING

These tests are designed to test the integrated software items to determine whether if they really execute as a single program or application. The testing is event driven and thus is concerned with the basic outcome of field. The Integration tests demonstrate that the components were individually satisfaction, as already represented by successful unit testing, the components are apt and fine. This type of testing is specially aimed to expose the issues that come-up by the components combination.

(III) FUNCTIONAL TESTING

The functional tests help in providing the systematic representation that functions tested are available and specified by technical requirement, documentation of the system and the user manual.

(IV) SYSTEM TESTING

System testing, as the name suggests, is the type of testing in which ensure that the software system meet the business requirements and aim. Testing of the configuration is taken place here to ensure predictable result and thus analysis of it. System testing is relied on the description of process and its flow, stressing on pre driven process and the points of integration.

(V) WHITE BOX TESTING

The white box testing is the type of testing in which the internal components of the system software is open and can be processed by the tester. It is therefore a complex type of testing process. All the data structure, components etc. are tested by the tester himself to find out a possible bug or error. It is used in situation in which the black box is incapable of finding out a bug. It is a complex type of testing which takes more time to get applied.

(VI) BLACK BOX TESTING

The black box testing is the type of testing in which the internal components of the software is hidden and only the input and output of the system is the key for the

tester to find out a bug. It is therefore a simple type of testing. A programmer with basic knowledge can also process this type of testing. It is less time consuming as compared to the white box testing. It is very successful for software which are less complex are straight-forward in nature. It is also less costly than white box testing.

(V) ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CHAPTER 6

CONCLUSION AND ENHANCEMENT

Conclusion

A case study of a flood scenario is simulated, through where Drone will investigate in clusters and give us a collective information, about the amount of damage caused by the disaster.

Future Work

In future we can replace the camara with thermal camara, to search people through the buildings in the disaster situation.

REFERENCES

1. "IEEE: 3D Optimal Surveillance Trajectory Planning for Multiple UAVs by Using Particle Swarm Optimization With Surveillance Area Priority ," NRF , HU TENG, ISHTIAQ AHMAD, ALAMGIR MSM , AND KYUNGHI CHANG Year:2020
 2. "Journal of Industrial Information Integration: Swarms of Unmanned Aerial Vehicles", Anam Tahira , Jari Bölingb , Mohammad-Hashem Haghbayana , Hannu T. Toivonenb , Juha Plosila Year:2019
 3. "IEEE: Energy-Efficient Formation Morphing for Collision Avoidance in a Swarm of Drones", JAWAD NAVEED YASIN , SHERIF ABDELMONEM SAYED MOHAMED , MOHAMMAD-HASHEM HAGHBAYAN , JUKKA HEIKKONEN , HANNU TENHUNEN , MUHAMMAD MEHBOOB YASIN , AND JUHA PLOSILA Year:2020
 4. "MDPI: Review of Unmanned Aerial Vehicle Swarm Communication Architectures and Routing Protocols", Xi Chen, Jun Tang Year:2020
 5. "MDPI: Beehive-Inspired Information Gathering with a Swarm of Autonomous Drones" Alberto Viseras , Thomas Wiedemann , Christoph Manss, Valentina Karolj, Dmitriy Shutin and Juan Marchal Year: 2019
-