Flying Agent: An Improvement to Urban Disaster Mitigation in RoboCup Rescue Simulation System

Pooya Deldar Gohardani, Peyman Ardestani, Siavash Mehrabi, MohammadAmin Yousefi

Mechatronics Research Laboratory, Islamic Azad University, Qazvin Branch, Qazvin, Iran

{Pooya.Deldar, Peyman.Ardestani}@Qiau.ac.ir
Siavash.Mehrabi@gmail.com

Abstract.
This paper introduces a new agent in RoboCup Rescue Simulation System to assist and improve search and rescue operation in urban areas. We define a new flying agent to provide air patrol and collect urban area information. Additional features of this agent can make this league more challenging, more interesting and help it to be closer to what happens in reality.

Keywords. RoboCup Rescue Simulation, Urban Search, Flying Agent

1 Introduction

Rescue Simulation [1] League has three kinds of agent; Ambulance, Police and Fire brigade. These agents are bound to move on the roads and within buildings. In scenarios with too many blocked roads and/or huge maps, another agent that can search the roads and collect information without getting stuck is very useful. Thus a new type of agents are defined as flying agents, they can fly over the urban areas without any trouble. Also they have distinct ability of imagery.

This type of agent improves the search and rescue process by its abilities but also adds some new challenges to the league. One of these challenges is to process the received images to perceive the environment information. In order to add this agent to current version of rescue simulation system[1], some changes have been done in various parts of the system. Some of these changes are such as move in three dimensions, imagery capabilities and new commands to handle the agent's actions. These changes are completely explained in different parts of this paper. We are going to present the flying agent, reasons to use it and its performance in next section. In third section we will look into differences between this new agent and other agents. In fourth section an instance of the new agent setting is presented. Finally conclusion and references are presented.

2 Flying Agent

In the following sections we will introduce the definition of a flying agent, why it is needed in areas such as Rescue Simulation System and how it operates and how it can be used considering its desired abilities.

2.1 What is a Flying Agent?

As mentioned before, Flying Agent is an agent with lots of abilities, which can affect search and rescue process widely. This kind of agents can be any types of agents from Helicopters to quad-rotors.

a) Helicopter b) Quad-rotor

Fig. 1.samples of flying agents

But they should have common features to be called as Flying Agents which are listed below:

- An agent with a variety of sensors
- An agent that can help other team agents (Ambulance team agents, Fire Brigade agents, Police Force agents)
- An agent which can fly over burning buildings, blocked roads and other rough environmental conditions
- An agent more effective in exploration missions

2.2 Why Flying Agent is needed?

Alongside different types of robots which are used in the process of search and rescue, flying robots play an important role, they can fly and gather information from a widespread area and help the ground rescue agents to act better during their rescue processes. Even they can incorporate directly at the rescue process for example by pouring water or other materials on the burning buildings. In addition to these abilities there are more advantages to use these types of agents, especially in Rescue Simulation Platform which can be as follows:

- Helping to fill the gap between Agent league and Virtual Robot league[2]

- To add more different challenges to the league

Flying agents are used alongside ground agents in Virtual Robot League for many years. These agents in Rescue Simulation System, which are large-scale version of virtual robots, were missing. It is tried to reduce distances between these two leagues by adding this type of agent. Also this new type of agent creates new challenges in the field of image processing. We hope to attract more people to this field by adding new challenges.

2.3 How it works?

In this section, main operations of a flying agent will be introduced, which will be descriptions on how it works, and what kind of constraints are in front of it. Operations of a flying agent can be categorized in four fields as movement, information gathering, communication and rescue. And there are two properties depending on its operations as *energy* and *health* which are described in the following sections:

Movement: It can move through three dimensions (x, y and z), so it can move over buildings and blocked roads which is impossible to other types of agents. Move action by following command: "AKFly (int time, List<Point3D> point3DList)" and agent will be able to move to specified point.

In Fig. 2, the agent's third dimension of movement is demonstrated. As shown in Fig. 2, if agent flies in low altitude the agent's marker would be smaller and if agent flies in higher altitude the agent's marker would be larger. It should be noted that flying agent has limitations in both minimum and maximum altitude for operation. In other words the lowest altitude for flying agents is bounded to buildings' height, and the highest altitude is determined based on parameters in flying agents' configuration file. The default value for flying agents' maximum altitude is 100000.

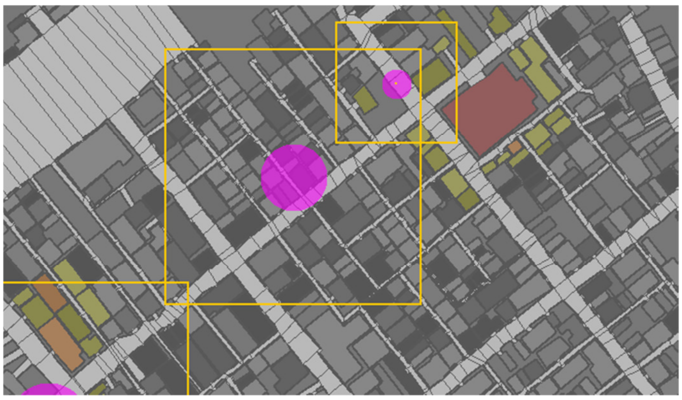


Fig. 2. Agent's third dimension of movement

Energy: There is a limited amount of energy for flying which means that after a while it should return to the refuge, recharge its power source and return to its tasks. Considering this feature, some works may be more energy consuming, for example when it flies in high altitude.

When a flying agent needs to recharge, it should land in a refuge. Thus two commands are added to these agents to provide takeoff and landing. These commands are *AKLand(EntityID agent, int time, EntityID landingZone)* and *AKTakeOff(EntityID agent, int time)*. More details are presented in further sections.

If agent runs out of energy before it reaches a refuge, the agent crashes and takes damage (conditions and amount of damage are explained in Health section). Taking damage will decrease the flying time of the agent. Based on [3] if the quad-rotor is used as the agent, according to eq.1 the consumed power will be such as below:

$$P = C_p \rho A (\omega r)^3 = \frac{T^{3/2}}{\sqrt{2\rho A}} \tag{1}$$

The eq.1 shows the used formula for flying agent energy consumption and the produced pressure in a rotor is shown in eq.2.

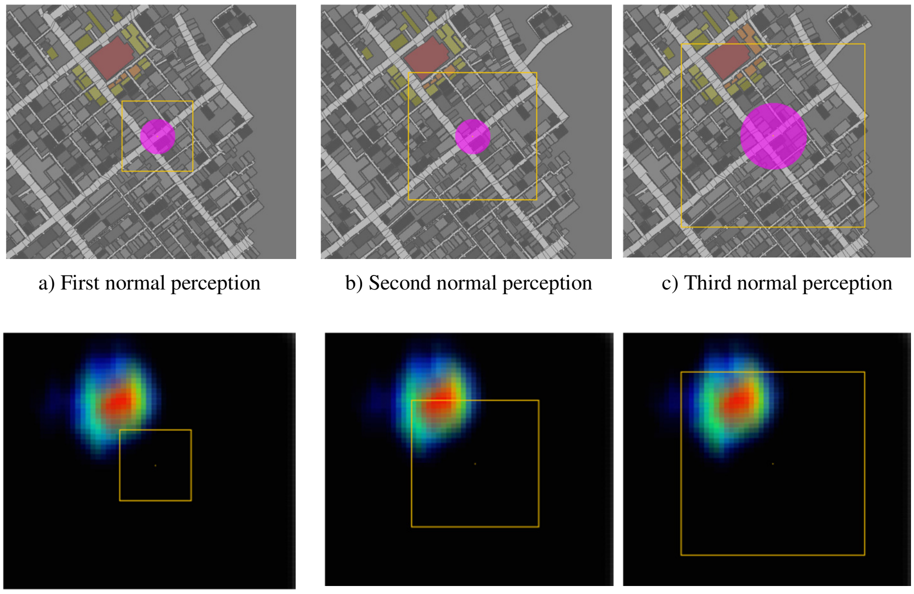
$$\mathrm{T} = C_T \rho \mathrm{A} (\omega r)^2 * f(HP) \tag{2}$$

$$\mathrm{f(HP)} = \frac{1}{1+e^{50-T/2}}\ ,\ T = \frac{HP}{100} (Health\ Precentage) \tag{3}$$

Where A is Area of single rotor disk, C_PCoefficient of power (0.008), $C_T$is Coefficient of thrust (0.009), P is Power consumed by a single rotor, T is Thrust produced by a single rotor,ρ is Air density (25°C,0m,75%humidity = 1.17kg/m3)[4] and ω is Angular velocity of the rotor.

Information Gathering. This kind of agent can have different facilities to gather information from its environment. These facilities can be categorized as imagery, olfactory or auditory. Considering the existing version of Rescue Simulation platform we have implemented imagery facilities and considered two types of imagery facilities in our implementation. We have implemented raw and infra-red imageries but it will not be limited to these. We are going to implement other types of imagery and other information gathering facilities.

Since this agent should capture some images from simulated environment and operate with images, another kind of perception was required. In Fig.3, some samples of agent perception are shown.



a) First normal perception   b) Second normal perception   c) Third normal perception

a') First infra-red perception   b') Second infra-red perception   c') Third infra-red perception

Fig. 3**.** Samples of agent perception

As you can see in Fig.3.a to Fig.3.c, agents' perception is shown as a square. Considering the agent movement along the vertical and horizontal axis, perception range would be different in different altitudes. If agent flies in higher altitude, more area will be covered but some noises would be added to images. Noise would be created by blur filter.

As shown in Fig.3.a' to Fig.3.c', another type of images is presented which are infrared. In this kind of imagery, areas with normal temperature are shown in black and when an area's temperature increases, the color varies from blue to deep red. In the same way when temperature decreases, color of area varies from deep red to black. This method of demonstration is based on fire simulator demonstration for temperature variation. Different types of imagery can be defined in configuration file.

Changeset is used in order to integrate image transfer to kernel and exchange these images between agent and kernel; like any other data transfer from kernel to agents. A property, named *ScannedImage* is added to flying agent that updates each cycle by kernel and images are available for agents by *ChanngeSet*.

Communication. This agent can communicate with other agents in the city through available channels. The agent takes images, processes them and collects data such as buildings on fire, blocked or opened roads then and then transmits them through its specified communication channel to proper receiver/s.

Rescue. Based on flying agent's description presented in previous section, this agent has many abilities in searching and data gathering. But in addition of these abilities, some of these agents can be utilized during rescue process too. Like an ambulance they can reach victims' locations and after conducting rescue operation, carry them to refuge directly by flying over the blocked roads and collapsed buildings. This ability needs a very strong flying agent and consequently it would consume lots of energy. So considering the other novelty that it brings into rescue simulation platform, we decided not to implement this feature for this presented work but we will implement it in near future.

Health. Just like other types of agents this agent has health properties which can be affected by some events. For example if it became too close to a burning building it may be damaged by the fire. Also if agent runs out of energy before it reaches to the refuge, agent will crash down and takes damage. The amount of damage varies based on temperature and impact intensity. The amount of damage is calculated by formula 4.

$$Dmg_{fall}(m) = \begin{cases} \frac{m}{n} * 9000 & if\ m < n \\ 10000 & if\ m \geq n \end{cases} \quad (4)$$

Where m shows the current altitude of the flying agent, n is the altitude that if agent falls from, it will be completely crashed, and the agent's stamina is 10000. This formula can be replaced by formula 5, so in case of falling from higher altitude than its threshold, the damage to this agent will be more than 10000.

$$Dmg_{fall}(m) = \frac{m}{n} * 10000 \quad (5)$$

To show temperature damage, formula 6 is presented, tmp_{hgt} is effective rate of temperature in current altitude for an agent. The amount of damage to the agent is calculated based on formula 6.

$$Dmg_{temp}(\text{tmp}, m) = \frac{1}{1 + e^{\frac{tmp_{hgt} - T}{5}}} * 10000\ ;\ T = maximum\ temprature/2 \quad (6)$$

In formula 7, m is the current altitude of flying agent, and based on this the effective altitude of agent will be obtained. Also n is the highest altitude of flying.

$$tmp_{hgt}(m) = \frac{m}{n} * tmp \quad (7)$$

Taking damage has two consequences on flying agent's performance. One is amount of noises in received images increases and the other is health reduction. If agent's health rate gets lower than HEALTH_THRESHOLD which is determined in configuration file, agent could not fly anymore. If agent crashes down or lands, then ambulance agents can Load them and carry them to the refuges as before.

3 Summary of new things and changes

3.1 commands

- AKFly(EntityID agent, int time, List<Point3D> point3DList)
  agent The ID of the agent issuing the command.
  time The time the command was issued.
  Point3DList The list of points that forms a path to fly
- AKLand(EntityID agent, int time, EntityID landingZone)
  agent The ID of the agent issuing the command.
  time The time the command was issued.
  landingZone the ID of the landing zone.
- AKTakeOff(EntityID agent, int time)
  agent The ID of the agent issuing the command.
  time The time the command was issued.

3.2 Configuration File

MAX_ALTITUDE : 100000
DISK_AREA : 15 cm^2
POWER_COEFFICIENT : 0.008
THRUST_COEFFICIENT : 0.009
AIR_DENSITY : (25°C,0m,75%humidity = 1.17kg/m3)[4]
ANGULAR_VELOCITY : 1
INFRARED_IMAGERY : false
HEALTH_THRESHOLD : 1000
MAX_HEIGHT : 10
MAX_TEMPERATURE : 900

4 Conclusion and Future Works

In this paper we have introduced a new agent as a Flying Agent which operates in rescue simulation platform. Reasons of using this agent, unique specifications of this agent and differences between this agent and other agents are mentioned. The performance of the agent is explained in detail. Also some instances from agent's image output with their specifications are presented. In addition the limits of the flying agent implementation are discussed. As this agent is a new agent we have implemented just some prominent features and some features such as olfactory or auditory abilities and rescue abilities are ignored.

We hope this agent find its real place in search and rescue operation in rescue simulation platform.

References

1. *The RoboCup-Rescue project: a robotic approach to the disaster mitigation problem.* Tadokoro, S, Kitano, H and Takahashi, T. San Francisco : IEEE, 2000.
2. Jijun, Wang. *USARSim, A Game-based Simulation of the NIST Reference Arenas.* 2003.
3. Hoffman, Gabriel M, et al., et al. *Quad-roro Helicopter Flight Dynamics and Control: Theory and Experiment.* s.l. : American institue of aeronautics and astrimaustics.
4. air density calculator. *denysschen.com.* [Online] http://www.denysschen.com/catalogue/density.aspx.