GENERATING EVACUATION ROUTES BY USING DRONE SYSTEM AND IMAGE ANALYSIS TO TRACK PEDESTRIAN AND SCAN THE AREA AFTER DISASTER OCCURRENCE

Aljehani Maher[1] and Masahiro Inoue[2]

Graduate School of Engineering and Science, Shibaura Institute of Technology

Saitama Japan

E-mail: mf14502@shibaura-it.ac.jp[1], inouem@shibaura-it.ac.jp[2]

ABSTRACT Throughout the ages natural disasters (i.e. earthquake, tsunami and eruption) are unavoidable casualties in the history of human being. Figure 1 demonstrates the percentages of people killed by different natural disasters. In many incidents after the earthquakes people died due to unplanned evacuation procedures. Data like status of evacuation routes and real-time alert system are required to be available after disaster occurrence. Implementing Drones in this area of study is so beneficial for many reasons. For instance, Drone has a high speed of covering the route's status, constructions' conditions and pedestrian movements aerially and visually at real time. In this study we focus on integrating Drone in M2M system as an end device.

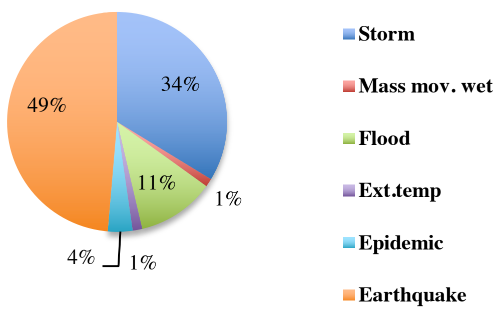

Fig.1 Chart of people killed by natural disasters

1. INTRODUCTION

Tsunami and earthquakes destroy houses, properties, and so on in the coastal and drifts them. Japan has learned lessons from the tsunami that occurred in 2011. Kusano & Inoue (2013) proposed a research called Safety Route Guidance System using Participatory Sensing. In many occasions map data cannot be used due to the environment status and the extreme affection to the area after the disaster occurrence. Also, there is a highly possibility of the default evacuation routes can't be safe to be taken anymore due to change of the geo-structural environments. The Safety Route Guidance System has a capability to provide an alternative map based on gathering sensing data from pedestrians' mobile phones with GPS and accelerometers on the disaster area. The main goal from that research is how to make alternative map and evaluate the routes in case of devastating earthquake and tsunami occurred.

The only inadequacy at that study is the concept of implementing mobile as an end device and data source. The idea of using alternative end device instead of the mobiles and sharing the information with other devices is more reliable than using only participator sensing system alone.

1. 1 Purposes and Motivations

Handset isn't a reliable data source since the functionality is consecutive to the pedestrians' holding availabilities and their behaviors. Not to mention the environmental data and geo-photography structural damages information after the disaster are essential visual data to reveal a reasonable evaluation of the new alternative routes in many critical situations like floods for instance. However, usually refugees are so dispersed and distracted after the disaster occurrence, so unfortunately visual data cannot be acquired by using handsets without refugee's contributions. Therefore, we implemented Drones and employing its properties as an end device and data source instead of handsets system.

This system contributes to generate an alternative evacuation maps semi-autonomously and individually after the earthquakes occurrence in short period of time. Also, we can take advantages of integrating image processing to track predestines autonomously and scan the impacted area.

1. 2 Proposed System

Recently Drones attracted a lot of attentions in different field of studies. In current usage, the acronym of UAV refers to an aerial vehicle with no human onboard. Initially, Drone was made to be a war flight for executing military operations remotely. It can carry missiles and apply reconnaissance of targeted areas. However, in our study we implemented Drones as a 3rd party robot or as an end device of M2M architecture to track pedestrians, create routes and provide visual data of the damaged area after disaster occurrence. Figure 2 illustrates our proposed system.

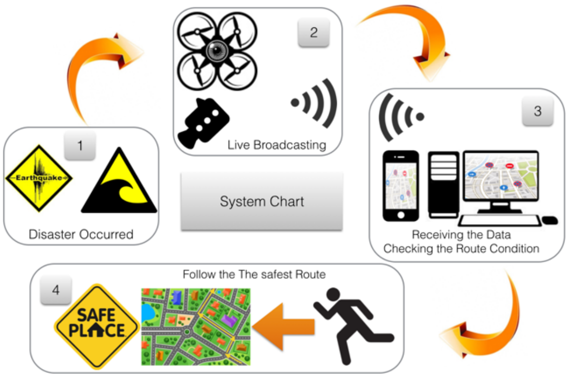

Fig.2 Proposed system

2. DRONE SELECTION

In the beginning we need to decide which Drone model we should implement as a tracker and broadcaster in this system, after that, we have to check the performance and the capabilities of selected Drone. Table 1 is a comparison table we have made between the most common Drones in the market.

Table 1 Drones comparison

Drone Model	Flying Time	Charging Time
DJ Phantom II	20-17(min)	180(min)
DJI Inspire	25-20(min)	180(min)
Parrot Bebop	11-8(min)	90(min)
TBS GEMINI	10-7(min)	100(min)
Quanum Nova	15-12(min)	90(min)
AR Parrot 2.0	18-12(min)	90(min)

We found AR. Drone is the suitable Drone that fulfills the proposed system requirements for the following reasons.

- HD Camera. 720p 30 fps and 60 fps vertical QVGA camera for ground speed measurement.
- Open source and programmable system
- Fully reprogrammable motor controller, Water resistant motor's electronic controller.
- Ultrasonic sensors for ground altitude measurement.

AR. Parrot 2.0 is an open source UAV. It can be modified and controlled by multiples applications. We used the original SDK of the Drone's API then we integrated OpenCV library to perform image processing and tracking part. Then, we made experiments to follow targets and people with different ways to see how the Drone tracks and provides data.

3. EXPERIMENTS

The system has different processes for providing the evacuation data after the disaster. Figure 3 demonstrates the system flowchart. There are tracking process and reconnaissance and surveillance process to provide the evacuation map. Each of them works semi autonomously.

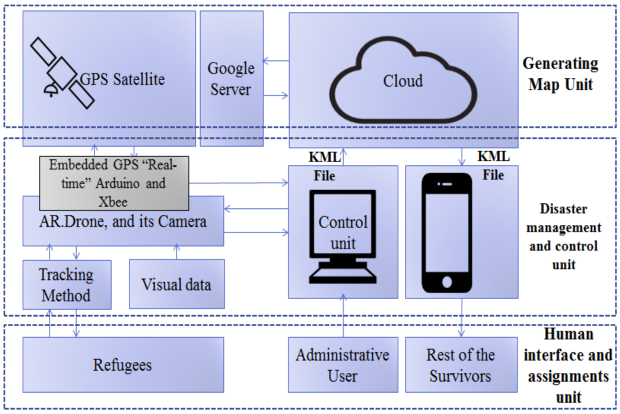

Fig.3 System flowchart

3.1 Tracking Process

After the disaster occurrence the system will initialize the tracking process, which can be executed if there are pedestrians on the damaged area. The tracking process can be processed by multiple ways of algorithms. We used OpenCV library to control the movements of the Drone according to the target position on the area. Then system will record the movements of the Drone by embedded G.P.S. on the board.

3.2 Reconnaissance and Surveillance

This process scans the area and confirms the reliability of the extracted routes. On the other hand, all Drone modules can be implemented on the proposed system. And each Drone has its own way of programming method to track or fly autonomously. Fortunately, AR. Drone has been made and prepared to track. And that made it the first choice to the large group of researchers from different fields of engineering.

3.3 Experimental Apparatus

In experimental apparatus we need AR. Drone and control unit. Usually control unit consists of computer, Arduino and G.P.S. shield. We used Arduino to generate the route according to the movements of the Drone. To control AR. Drone, we need to deal with AT commands. AT commands are encoded as 8-bit ASCII characters with a carriage return. All AT commands start with "AT*" followed by a command name, an equals sign, a sequence number. The reference of all the available AT commands are listed below, it is available in the AR. Drone open API documentation made by Stephane, P & Nicolas, B AR. Drone Developer Guide (2011).

- AT*REF (input) - Takeoff/Landing/Emergency stop command
- AT*PCMD (flag, roll, pitch, gaz, yaw) - Move the drone
- AT*PCMD_MAG (flag, roll, pitch, gaz, yaw, psi, psi accuracy) - Move the drone (With Absolute Control support)
- AT*FTRIM - Sets the reference for the horizontal plane (The drone must be on the ground)
- AT*CONFIG (key, value) - Configuration of the AR. Drone 2
- AT*COMWDG - Reset the communication watchdog

To control the AR. Drone, we need AT*PCMD for moving process and AT*REF for taking off and landing. Table 2 shows the commands values.

Table 2 Standard AT commands of AR. Drone

AT*Flag	Corresponding AT commands value range
AT*REF	Integer value = 1 Takeoff, 0 Land
AT*PCMD_MAG	Float (Left/Right) angle ∈ [–1.0; +1.0]
AT*PCMD_MAG	Float (Front/Back) angle ∈ [–1.0; +1.0]
AT*PCMD	Float Vertical speed ∈ [–1.0; +1.0]
AT*PCMD	Float Angular (yaw) ∈ [–1.0; +1.0]

3.4 Technique

Firstly, we need to build Parrot SDK of AR. Drone 2.0. In the control unit, we used SDK 2.0.1 on Ubuntu 14.04 LTS with latest firmware 2.4.8 of AR. Drone 2.0. We run some examples to check if the SDK is installed properly. After that, we have to install specific packages of OpenCV. It must be installed individually. Table 3 shows the OpenCV packages that must be installed.

In order to get the video streaming from AR. Drone and use it in OpenCV, we have to disable the GTK. GTK is the default video streaming in AR. Drone SDK2.0.1. In order to make OpenCV the default application and responsible for images processing and video streaming, we must edit some scripts in the original SDK packages before building the file on the system.

After that, we have to change the format of broadcasting from PIX_FMT_RGB565 to PIX_FMT_RGB24, since the OpenCV works well on RGB24 format. Then we added the function to convert the video frames to OpenCV image objects. Also, we need ROS, "Robot Operating System", which is a software framework that enables the development of robotic applications control and interact with AR. Drone.

Table 3 OpenCV packages

Package name	Function
libopencv-dev	OpenCV main library
libopencv-core	Contains the header files and static
libopencv-highgui	High-level GUI and Media I/O library
libopencv-highgui-dev	High-level GUI and Media I/O library.
libcv-dev	Translate from libcv-dev to subdivided packages.

3.5 Tracking procedure

Experimentally we were trying to follow a targets using multicolor feature. By adjusting Hue, Saturation Brightness values. These three factors have scale of 0-256. We can train AR. Drone to trace the target according to its color. Then, we trained Drone to track the upper body and the full body. The movements in these experiments were forward, backward, left and right.

4. TRACKING RESULTS AND PERFORMANCE

Firstly, we've tested the performance of AR. Drone's camera. To do so we tracked different targets which are yellow object, red object and human face. Then we compared the accuracy of detection according to the distance between the target and the AR. Drone camera. We used this equation for evaluation the accuracy.

$$[(F_d+12)/(F_a)] *100 = \% \textit{ Accuracy of Tracking}$$

- The respond time is 12 frames delayed.

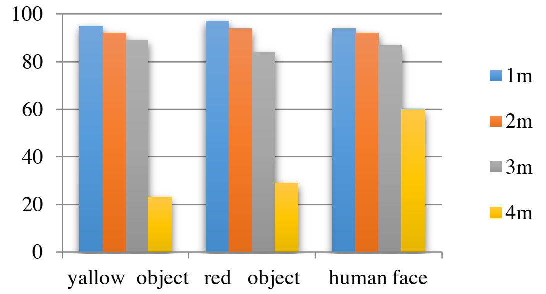

Fig.4 Result of AR. Drone camera performance.

4.1 Multicolor tracking process

To implement multicolor tracking, all we have to do is defining a list of boundaries in the RGB color space (or rather, BGR, since OpenCV represents images as NumPy arrays in reverse order), where each entry in the list is a tuple with two values, since these are pixel values that fall within the range of [0; 256] we can use unsigned 8-bit integer data type to track. Figure 5 shows the multicolor tracking scenario.

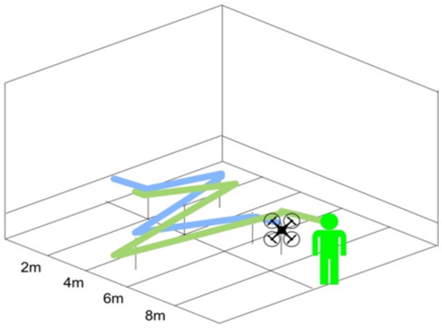

Fig.5 Multicolor tracking scenario

4.2 Upper Body tracking process

By using Haar based detectors to detect the upper part of the body. This method is classified as the robust pedestrian detection. This detector has been successfully applied to pedestrian detection in recorded video format and real-time streaming format. In case of using real-time streaming, upper body tracking can be achieved by implementing HaarFaceDetector parameters to the detector. Basically, after detecting the face's position, detector will estimate the upper body position. Figure 6 shows upper body tracking scenario.

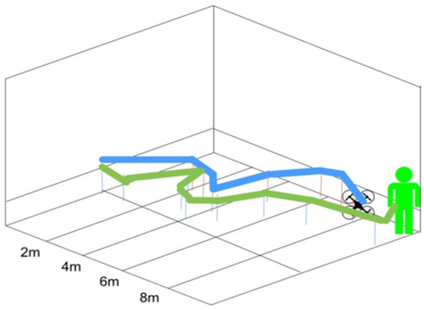

Fig.6 Upper body tracking scenario

4.3 Full Body tracking process

Unlike multicolor and upper body tracking, full body tracking is much more difficult to implement by using OpenCV haar detectors. Since, human face has a nice structure, which is more suitable for haar detector. But, full body is complex and background scene can greatly affect the tracking process. Figure 7 shows full body tracking, AR. Drone lost tracking in this technique.

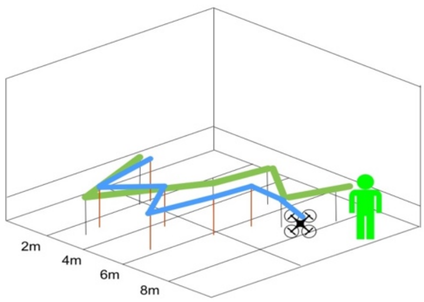

Fig.7 Full body tracking scenario

5. CONCLUSION

In this study we implemented AR. Drone as an end device to track pedestrians and provide the visual data after disaster occurrence. Drone is a reliable end device since it cannot be affected by disasters impacts. We made experiments on the AR. Parrot Drone 2.0 camera performance to check the reliability of tracking. Then, we made three different experiments for tracking procedures. According to the results of experiments we found upper-body tracking and multicolor tracking better than full body tracking. The tracking depends on the target appearances, environments characteristics and the tracking method. Therefore, we recommend human motoring to get an optimal tracking process.

NOMENCLATURE

F_a : Frame of Appearing.
F_d : Frame of Detection.

REFERENCES

Earthquake Data and statistics Data related to human and economic losses from disasters that have occurred between 1980 and 2008.

Kusano, S., & Inoue M., Safety route guidance system using participatory sensing, IEEE 2nd Global Conference on Consumer Electronics (GCCE) 2013.

Hannes, K., & Modesto, C., & Bernt, S., Fast and Robust Face Finding via Local Context Joint, IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking 2003.

Stephane P., Nicolas B., AR. Drone Developer Guide, SDK 1.6. 2011.

Eng. Maher Aljehani received his Associate Degree in Telecom from Jeddah College of Telecom & Electronics and B. Eng. degree in Telecommunication Engineering from Huazhong University of Science and Technology. Since 2014 He has been a master student at Shibaura Institute of Technology.

Dr. Masahiro Inoue received his B. S. and M. S. degree in Applied Physics from Waseda University and Ph. D degree in Computer Science from Shizuoka University. He was engaged in research and development at Mitsubishi Electric Corporation. Since 2005 he has been a professor of Shibaura Institute of Technology. He is a Professional Engineer in Japan and a Project Management Professional.