Drone Monitoring System for Disaster Areas *

Extended Abstract[†]

Patrick Hock
Faculty of Science and Engineering,
Saga University
Saga, Japan
16233064@edu.cc.saga-u.ac.jp

Koki Wakiyama
Graduate School of Science and
Engineering, Saga University
Saga, Japan
19704023@edu.cc.saga-u.ac.jp

Chika Oshima
Faculty of Science and Engineering,
Saga University
Saga, Japan
karin27@sa3.so-net.ne.jp

Koichi Nakayama
Graduate School of Science and
Engineering, Saga University
Saga, Japan
knakayama@is.saga-u.ac.jp

ABSTRACT

In this paper, four systems for disaster countermeasures were implemented. The first one was a 3D reconstruction system. It would construct a 3D model from images taken by a drone. The second one was a human detection and posture analyzing system. A drone would find a person in eyesight, take a picture and let it be posture analyzed. This would help finding people in need. The third one was a self flying system for drones. Drones would be given a mission to carry out. Real time analysis of their eyesight would allow autonomous decisions for flying routes. The fourth one was a system, where one server would control multiple drones at the same time to gather information about remote disaster areas. In an experiment, the AI drone flew autonomously and created useful data for monitoring the area. As multiple drones could be used at the same time, this system has a large potential for making disaster area monitoring easier and disaster countermeasures more efficient.[1]

CCS CONCEPTS

• **Computing methodologies → Artificial intelligence → Planning and scheduling**

KEYWORDS

AI drone, image processing, rescue support, system, disaster observation

ACM Reference format:

P. Hock, K. Wakiyama, C. Oshima and K. Nakayama. 2019. SIG Proceedings Paper in word Format. In *Proceedings of ACM GECCO conference, Prague, Czech Republic, July 2019 (GECCO'19)*, 5 pages.
DOI: 10.1145/3319619.3326885

1 INTRODUCTION

Disasters are ever-present – natural or man made. If they happen, it might become difficult to gather information about the current on-site situation, due to obstructed passages. Even if it's possible, it might take a comparably large amount of time to reach the area. An observation from the air would be a desirable solution for this problem.

In this paper we present the first steps for a system, that enables monitoring of hard-to-reach areas using drones. The drones are free from any transmission distance limits. Given a target location, the drone will move there and can

- find people in eyesight
- shoot a single picture for person posture analysis
- shoot a group of pictures for 3D reconstruction.

2 3D RECONSTRUCTION SYSTEM

It's possible to reconstruct a scene as 3D model using pictures or videos shot by the camera of a drone.

For the 3D reconstruction process we use the photogrammetry software *Pix4D mapper* (*Pix4D*).

To create a high quality 3D model it's necessary to provide a certain number of images with a certain overlap.

2.1 Picture Shooting

[1]Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic
© 2019 Copyright is held by the owner/author(s).
Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3326885

Figure 1 shows the path and image count necessary to gain a precise 3D reconstruction. The green triangle shows the starting point and moving direction of the drone. Each red dot implies a photoshoot. The blue cross in the middle describes the point of interest. All pictures have a total overlapping area of 90%.

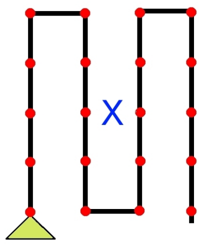


Figure 1: 90% overlap, 20 image M-path

2.1.1 Overlap criteria. The general criteria for picture overlapping are as follows:

- Front overlap: more than 75% (Figure 2)
- Side overlap: more than 60% (Figure 2)

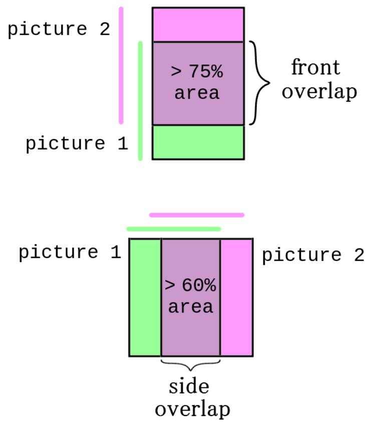


Figure 2: Front overlap, Side Overlap

2.1.2 Image Count and Path. Our experiments have shown, that an image count of 20 images and an "*M"* shaped path around the point of interest (Figure 1) creates a useful 3D reconstruction.

In favor of 3D reconstruction processing speed we also experimented with an image count of 12 and a respectively shorter "*N"* shaped path (Figure 3) and a circular path. However, both methods did not deliver satisfying results (see Figure 5).

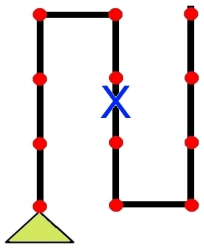


Figure 3: 90% overlap, 12 image N-path

2.2 3D Reconstruction Results

The results of our 3D reconstruction test is shown in Figure 3 and Figure 4. For this reconstruction test we used the following setup.

- Drone: DJI Phantom 4
- Camera: built-in camera of drone
- Drone flying height: 25m

The 20 image M-path resulted in Figure 4. The photo shooting process took 1:56 minutes. The 12 images N-path resulted in Figure 5. The photo shooting process took 1:27 minutes.

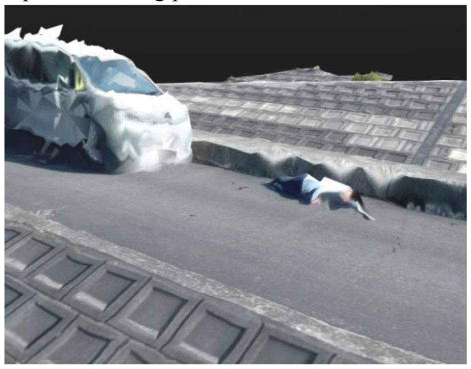


Figure 4: 3D reconstruction from 20 image M-path

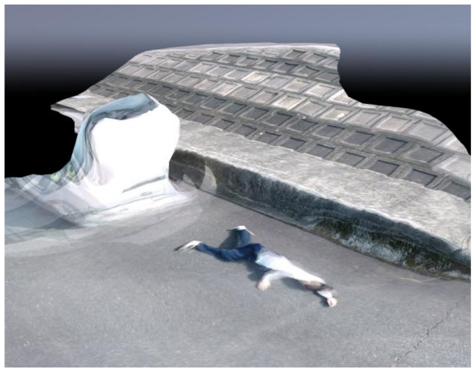


Figure 5: 3D reconstruction from 12 image N-path

3 HUMAN POSTURE ANALYZING SYSTEM

This system is made to analyze the posture of a person. We aim to determine if she/he is in need of help. For this analyzation, we use OpenPose [4]. OpenPose can recognize people on images and calculates their estimated bone structure. This reveals their posture and allows an estimation of how much help they might need.

3.1 Workflow

The workflow from "image taken" to "present processed image to user" is as follows:

- Drone takes a picture with its built-in camera.
- Picture is transferred to the Android tablet via the remote control of the drone.
- Android tablet transfers image to a server using HTTP.
- Server transfers image to a high-spec server.
- High-spec server calculates and overlays bone structure.

- High-spec server sends the result back to usual server.
- Server adds resulting image to the image gallery of results

3.2 Experiment Results

For our experiment we used a DJI Inspire1 drone with a DJI Zenmuse X3 camera attached to it. We manually steered the drone to a certain position, took the pictures and let them be processed. Figure 6 shows the raw picture. Figure 7 shows the processed picture.

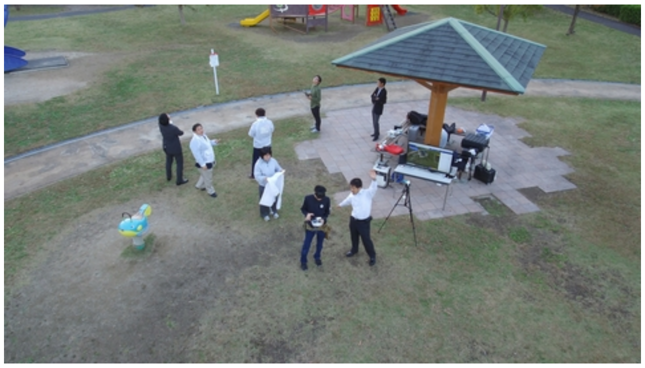


Figure 6: Raw image taken by drone.

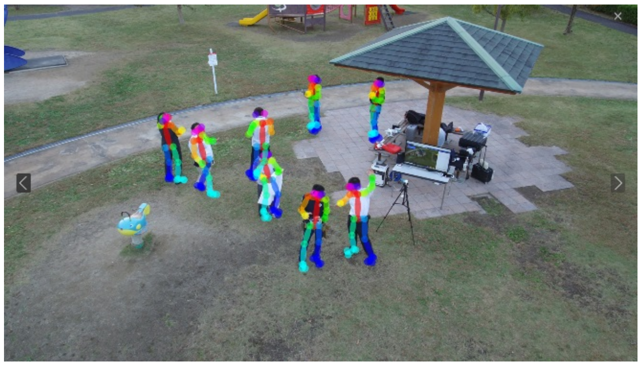


Figure 7: Image with bone structure processed with OpenPose.

3.3 Presentation of Results

Resulting images with bone structure overlay are provided in an image gallery using HTML and Javascript (baguetteBox library [5]) (Figure 8).

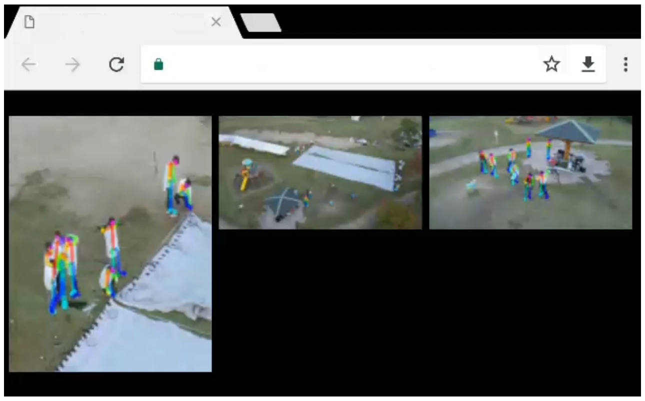


Figure 8: Gallery of OpenPose bone structure result images.

4 SELF FLYING SYSTEM (USING RASPBERRY PI)

Usually a drone is bound to a transmission distance limit between drone its remote control. The self flying system overcomes this transmission limit, so the drone is only limited by its battery life. For a Phantom 4 this enables a flight distance of roughly 35km (without return) under optimal circumstances.

To enable computer vision the drone the system uses OpenCV [7] as just-in-time image processing library.

The connection between Raspberry Pi and the drone is achieved through the DJI Onboard SDK.

A drone with a Raspberry Pi attached to it is henceforth called *AI drone*.

4.1 DJI Onboard SDK

The DJI *Onboard SDK* is a *C++* library for creating computer programs that can control the flight course and camera of a drone.

The hardware connection on the drone is realized through a simple serial TX/RX interface.

4.2 Raspberry Pi

The Raspberry Pi is a lightweight single-board computer developed by the Raspberry Pi Foundation. It's using an ARM processor and a (micro)SD card as storage. It offers various extension possibilities, such as a camera.

For our experiments we used the following configuration:

- Model: Raspberri Pi 3 Model B+
- OS: Raspberian OS (Kernel Version 4.9)
- Drone SDK: DJI Onboard SDK 3.6
- Computer vision: OpenCV
- Camera: (Official) Camera Module V2

4.3 Camera Module of Raspberry Pi

The DJI Onboard SDK does not provide a limitless access to the built-in camera of the drone. Therefore we equipped the Raspberry Pi with its *camera module V2* to enable computer vision.

The basic interface for the Raspberry Pi camera is Python or the command line. The DJI Onboard SDK however works on

C++. To find a common ground for both and enable the use of OpenCV, we used the library RaspiCam_CV [8].

4.4 OpenCV

OpenCV is an open source library provided by Intel. We use OpenCV to enable just-in-time image analysis with the AI drone. This analysis affects the flight behavior of the AI drone.

4.5 Hardware Assembly

The Raspberry Pi is attached on top of the drone. It connects via a USB UART adapter from its USB port to the serial port or the drone. The Raspberry Pi camera is connected via its dedicated connection cable. (Figure 9).

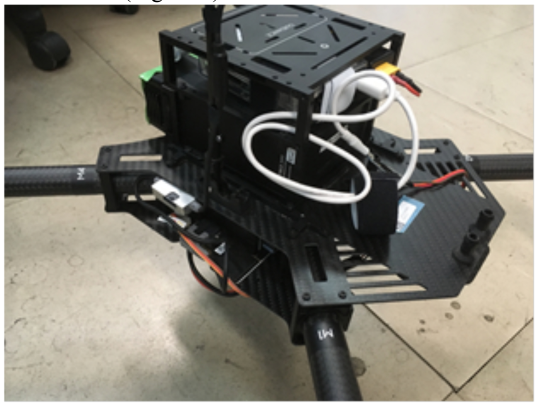


Figure 9: Raspberry Pi attached on top of drone

4.6 Controlling AI Drones

AI drones are controlled by transferring a *mission* their Raspberry Pi. A mission is a datatype containing the following information:

- Drone ID
- Mission type (E.g. "make a 20 image M-path shot")
- Target location (including height)

5 DRONE MONITORING SYSTEM FOR DISASTER AREAS

Combining the above described systems we create a drone monitoring system for disaster areas. With this system we are able to monitor areas using multiple drones at once, that are not depended on a remote controller's transmission distance.

5.1 Workflow

The workflow for the usage of the drone monitoring system is as follows:

- AI drones are awaiting new missions.
- Human user creates a new mission via HTML interface.
- Server sends the mission to the specified AI drone.
- AI drone moves to specified location.
- AI drone uses cameras and OpenCV to find an object of interest (e.g. a human being).
- AI drone moves above the object of interest and takes another picture.
- AI drone returns to base.
- AI drone awaits a new mission.

5.2 Structure

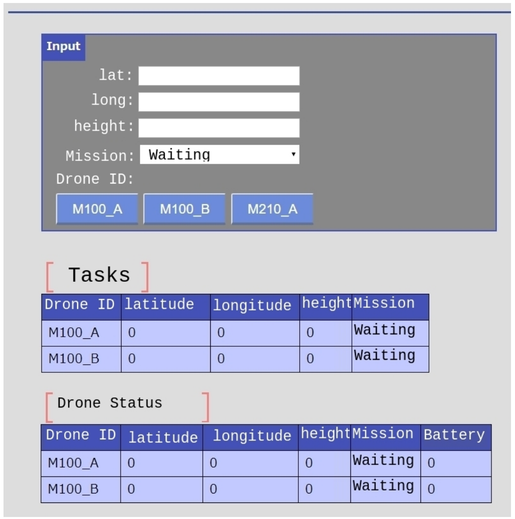
The system is built to enable controlling of multiple AI drones at once.

All AI drone statuses and missions are stored on a dedicated database server. This enables the following functionality for the future:

- It's easy to share information with different platforms. This can avoid drone-on-drone crashes and might enable mission collaboration.
- Given the mission parameters (without a drone ID), the server could autonomously choose the most suitable AI drone for it.

5.2.1 Server. The server provides an HTML control panel interface. This enables a human user to create missions for AI drones.

Figure 10 shows a screenshot of the control panel. New missions are created under "Input". Active missions are listed under "Tasks". Current statuses of AI drones are listed under "Drone Status".



lat:
long:
height:
Mission: Waiting
Drone ID:

Tasks

Drone ID	latitude	longitude	height	Mission
M100_A	0	0	0	Waiting
M100_B	0	0	0	Waiting

Drone Status

Drone ID	latitude	longitude	height	Mission	Battery
M100_A	0	0	0	Waiting	0
M100_B	0	0	0	Waiting	0

Figure 10: Screenshot of drone monitoring system.

The control panel is implemented using PHP and MySQL.

Mission data and drone statuses are stored inside a MySQL database. Updating mission data is accomplished by simply overwriting the current value inside the database.

Every issue or update of a mission also gets logged in a separate log file.

5.2.2 AI Drone. AI drones are implemented using the self flying system as a foundation. Mission issuing is done through the server using HTTP.

While the AI drone is in *waiting* mode, it frequently (every 5 seconds) asks the server for a mission. If the AI drone detected a new mission or a mission update, it carries it out. After that, it goes back into waiting mode to await another mission.

To connect the Raspberry Pi with the server, we're currently using WiFi. This is due to a current Japanese law restriction, which does not allow 4G devices to be used in mid-air.

5.3 Data Presentation

All communication data between AI drone and server are being stored. Therefore it's possible to visualize the data on a map using a GPS visualization API like Google Maps [9]. This allows the visualization of the flight path of an AI drone.

6 FURTHER APPLIANCE: MEDICINE TRANSPORT

As with on-site monitoring it might be difficult to deliver medicine to disaster areas as well. Therefore we also tested, if the monitoring system could deliver light-weight medicine. In our Experiment we assumed the transport of insulin for diabetes patients.

6.1 Experiment

We attached a box of fluid lipstick as *fake insulin* onto an drone using cellophane tape. We then created a waypoint mission to let the drone fly from "Nishi-taku" to a small river in about 4km distance (Figure 11).

This experiment was not carried out using the self flying system / monitoring system described above, but with an Android application and the drone's remote control. This required us keep a connection between the drone and controller. Therefore we followed it by car.

At its destination the drone was hovering at a height of 0.3m. This enabled detaching the box with the *fake insulin*. We then let the drone return back to its starting point.

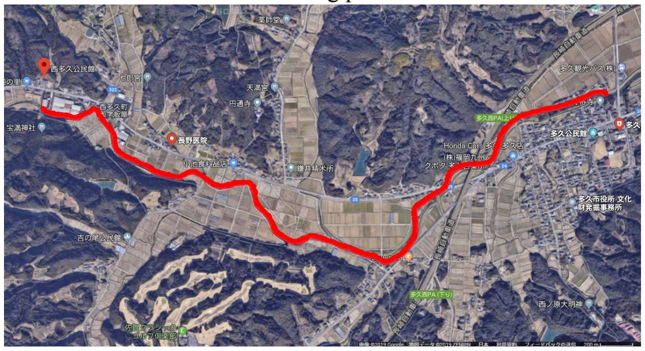


Figure 11: Flight route of medicine transport drone.

6.1 Experiment Results

The experiment showed, that its possible to let a drone carry light-weight medicine to a remote location.

In the future the medicine transport can be easily integrated into the monitoring system.

7 CONCLUSIONS

In this research we successfully implemented an AI drone, that is able to autonomously carry out certain tasks.

We were able to find and analyze the posture of people and create 3D models of specified locations. This enables us to monitor disaster areas even from remote locations.

However, not all stand-alone systems are yet integrated into the final drone monitoring system for disaster areas. Therefore, in its current state, we are only able to monitor local locations.

After the integration of all stand-alone systems we will be able to monitor various disaster areas in different locations at the same time.

ACKNOWLEDGMENTS

This work was supported by TAKATA foundation H30 and JST-Mirai Program,17943904

REFERENCES

[1] Mr. Nakamura 2017. Using Unmanned Aerial Vehicle（UAV）for disaster coping by Geographical Survey Institute, Hokkaido messurement technique tecture, 2017: http://www.gsi.go.jp/common/000150883.pdf (Japanese)

[2] DJI Mobile SDK Documentation: https://developer.dji.com/mobile-sdk/documentation/introduction/index.html

[3] Pix4Dmapper Image acquisition: https://support.pix4d.com/hc/en-us/articles/115002471546-Image-acquisition

[4] OpenPose: https://github.com/CMU-Perceptual-Computing-Lab/openpose

[5] baguetteBox: https://github.com/feimosi/baguetteBox.js

[6] DJI Onboard SDK Documentation: https://developer.dji.com/onboard-sdk/documentation/introduction/homepage.html

[7] OpenCV: https://opencv.org/opencv-4-0-0.html

[8] RaspiCam_CV: https://github.com/robidouille/robidouille/tree/master/raspicam_cv

[9] Google Maps API: https://cloud.google.com/maps-platform/?hl=ja