Post-Disaster Recovery by Rapid High Resolution Aerial Imaging and Indoor 3D Mapping with Autonomous Quadrotor UAVs driven by Computer Vision Feature Targeting and Real-time Victim Recognition

1 Introduction

Using our sub-$500 unmanned aerial vehicles (UAVs), we can save thousands of victims of floods, earthquakes, hurricanes and other disasters with faster and more effective response. We created a hardware and software platform to produce aerial maps for first responders. 3D indoor models are sent to remote doctors for victim diagnosis. Victims are automatically detected so rescuers can be quickly dispatched to people in need.

Figure 1: Flying quadrotor UAV[1]

Figure 2: Mounted phone imaging unit

1.1 Brief

1.1.1 Project Components

Autonomous Quadrotor UAV for Aerial Imaging progressively captures up to $5cm$ resolution aerial imagery suitable for facial recognition for under $\$500$

Agile Quadrotor for Indoor 3D Mapping maps the interiors of buildings too dangerous to enter with a consumer level 3D camera

Central Image Processing Server that all drones communicate with. *Stitches images*, *assembles 3D point clouds*, and *controls the drone network*

Rapid multi-pass facial recognition can process gigabytes of high resolution imagery to locate victims performantly

1.1.2 Problem: Aerial Mapping

During Hurricane Sandy and the Boston bombings, first responders used commercial level aerial imagery to coordinate search and rescue efforts.[2] Aerial imagery is **critical to disaster response—responders can prioritize recovery efforts with increased situational awareness**. High resolution aerial imagery thereby maximizes rescue resources and helps to **save more lives**. By far, the most commonly used tool currently is Google Earth, however with extremely low resolutions ranging from 1m to 15m making ***current imagery inadequate***.[3]

Even with high resolution maps, responders need immediate, up-to-date maps to locate damaged areas; this requirement makes any mapping technology that is not real-time or rapidly deployable inapplicable. With current imagery and mapping technology, **automated search for survivors is infeasible**.

1.1.3 Problem: Indoor Victim Search and Rescue

First responders risk their lives when entering buildings to search for victims. In situations where the disaster involves hazardous agents, like the Fukushima Nuclear Disaster, **searching for victims may not be possible** given the risk to responders.

1.1.4 Potential Application: Fully Integrated Automated Victim Search in a Disaster

With the high resolution aerial imaging platform developed in this project, **automatic search for victims has been created, freeing disaster responders to rescue identified targets while removing the need to search indoors. Preliminary results can be available within 15 minutes, upon which a UAV autonomously deploys, images a target and returns**.

1.2 Motivation - Potential Utility for Rapid Drone based Imagery

Fresh aerial imagery allows first responders to target highly damaged areas quickly and prioritize rescue efforts. By shifting the time first responders spend searching for victims to rescuing them, more lives can be saved, especially with time critical injuries such as blood loss.

1.3 Alternate Solutions

Unmanned Aerial Vehicles (UAVs) have occasionally been used in search and rescue efforts. However, current imaging approaches have many limitations. Existing models like the Mikrokopter and Aeryon Scout (**??**) require direct human control, and video captured needs to be analyzed by operators. Further, these solutions are cost prohibitive (Aeryon Scout is over $100,000). [4]

Satellites such as GeoEye-I can take 8-10 days to reach the disaster area, post-storm cloud fields can obstruct clear view, and non-military satellite imagery has a capped resolution at $\frac{1}{2}$ meter at best, often at a resolution as low as $15m^2$ per pixel. [5]

Commercial drones typically range from $\$10,000$ to over $\$100,000$ while providing little additional value for rescue teams as most lack autonomous capabilities, requiring a specially trained operator, making such drones cost-prohibitive for many organizations. [6]

1.4 Quadrotor Advantages

- Quadrotors are similar to helicopters but have more maneuverability meaning quadrotors can hover in place, extremely important for aerial photography.
- Diametrically opposite blades rotate in the same direction, with one set rotating clockwise and one rotating counter-clockwise.
- The counter rotation of adjacent blades results in no net torque on the quadrotor.
- VTOLs (Vertical Takeoff and Landing): Drones can launch and land in confined spaces on many terrains
- Reduced mechanical complexity means safer and lower cost than helicopters

1.5 Project Goals

Our engineering goal is the design and construction of an *affordable* autonomous quadrotor based aerial photography system for rapid acquisition of accurate and up-to-date maps to aid disaster response or commercial interests coupled with indoor 3D scanning to allow responders to locate victims in a damaged building.

1.6 Engineering Design Criteria

1. Function with daytime environments, winds under 20 km/h
2. Include GPS and other sensors for navigation
3. Individual unit less than $500
4. High resolution front and bottom cameras to ensure quality imagery.
5. The UAV should have an angular resolution better than 41cm to beat current satellite imaging technology

1.7 Team Contributions

Team Leader primarily focused on the construction of the quadrotor platform as well as the design and architecture of the distributed server cluster. This included the control systems that drove the quadrotor. Team Leader designed the 3D indoor iteration of the quadrotor which utilized a very heavily modified consumer RGB-D 3D sensor. Team Member created the image processing toolset and built the 3D stitching pipeline to generate point clouds for interiors of buildings.

2 Materials and Methods

2.1 High Level Process

1. A master quadrotor drone captures high altitude imagery of an area by automatically navigating between generated GPS waypoints.
2. Servers identify SIFT keypoints and stitch images together into map overlays.
3. Slave quadrotor drones are deployed to keypoint clusters and capture high resolution panoramas.
4. Indoor 3D maps are generated with drone mounted RGBD cameras.

2.2 AR.Drone

We modified the $300 Parrot AR.Drone 2.0 recreational quadrotor to serve as a base payload carrying platform. The lightweight drone is very inexpensive, although we had to strip much of the protective casing to increase additional cargo capacity. The UAV has a built in frontal camera, with a resolution of 720p (1280×720) and a 240p QVGA high frame rate bottom stabilization camera. The AR.Drone includes a logic board that coordinates the four rotors and stabilizes the craft with sensory input.

Summary of UAV specifications[7]:

- Barometer and ultrasonic sensor for altitude measurement
- Accelerometer and gyroscope
- 1280×720 pixel wide angle front camera
- 320×240 pixel QVGA high frame rate bottom camera for measuring ground speed

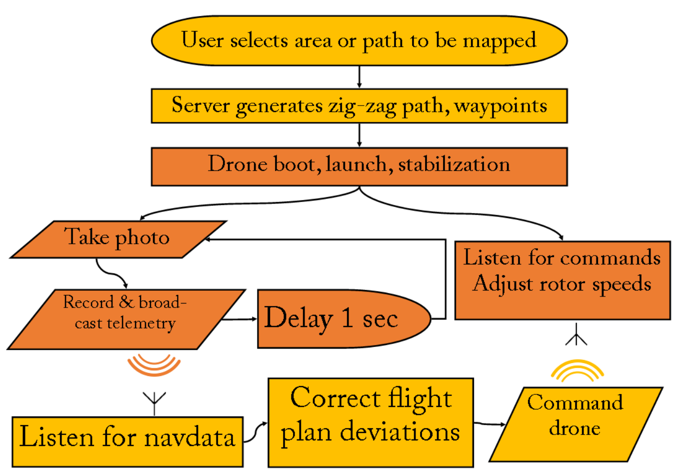
By using off the shelf hardware and modifying it for the project requirements, the price per drone can be kept low due to price savings from large scale mass production.

2.2.1 UAV Control

1. An operator draws a flight path in Google Earth and way-points are generated. (Figure 3)
2. The drone is launched and stabilized.
3. Photos and telemetry collected and transmitted to server.
4. Flight is adjusted when the drone exits a 3 meter wide corridor between way-points.
5. The flight plan is sent to the central server to be added as an imaging to the task queue for the drone fleet to deploy a charged UAV.

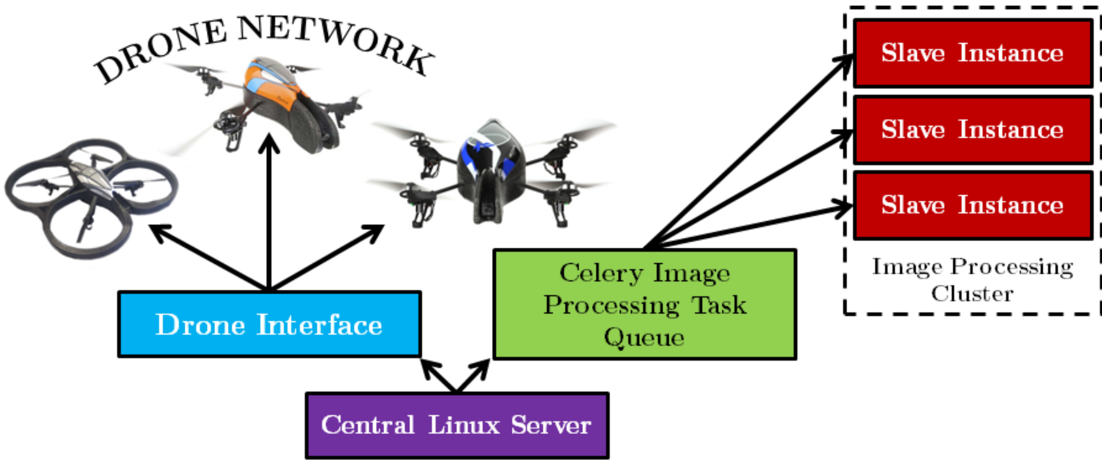
Figure 3: Google Earth flight plan

Figure 4: UAV control flow



2.3 Central Server

Figure 5: Server Cluster Architecture



A central Linux server acts as a central hub to which every drone connects to. This server handles drone management and acts as a load balancer to the rest of the cluster. As images are collected, it adds tasks to a Celery task queue. Server instances are dynamically created to complete the tasks in the Celery task queue. Such tasks include (but are not limited to) stitching, SIFT keypointing, locating faces and skin tone isolation.

Tasks are completed as a priority queue, allowing image processing jobs that are important (e.g. low resolution keypointing) to be completed first. This distributed system allows the system to become near-infinitely scalable by adding more workers. This process can be automated with a cloud service provider like the Amazon AWS Elastic Compute Cloud where new servers are dynamically provisioned as needed.

2.4 Phone Telemetry and Ground Imaging

Our final prototype used an inexpensive Android smartphone with a GPS sensor and a 3G data connection to record high resolution imagery and transmit telemetry. For development, we used the Nexus S, although cheaper and more lightweight devices could easily be substituted. With our *Drone Logger* Android application, an image from the back camera (facing downward when mounted on the UAV) are recorded every second to the phone's internal storage. Depending on the specific device used, different sets of telemetry are available; we captured and transmitted the **bearing, speed, latitude and longitude**.

Telemetry is sent to a local or remote server with a public IP address as an HTTP POST request. If telecommunications infrastructure is damaged and a data connection is unavailable on the phone, a portable server and the mounted phone would connect to an adhoc WIFI network. The AR.Drone creates an In the development of our prototypes, the smartphone, UAV creates a local network with an approximate range of 165 meters.

2.5 Prototypes

2.5.1 V1 – Drone bottom stabilization camera

Our initial prototype relied on the bottom facing stabilization camera bundled with the AR Drone. Using it for map creation would have minimized the overall costs, but the camera is of too low resolution (240p) for workable imagery.

2.5.2 V2 – Drone Mounted Arduino and Sensors (Figure 15)

To add cameras and additional sensors to the UAV, we mounted an Adruino microcontroller. This gave us the freedom to add additional sensors. However, the asymmetric weight distribution made the drone:

- Difficult to isolate issues and to use serial communication
- Unstable when mounted

2.5.3 V3 – Android Smartphone and PC Server

We overcame cost, data transmission and aerodynamic issues in previous prototypes by mounting an Android smartphone on the bottom of the UAV. The phone has several benefit to a custom Adruino based system:

1. Even low-end phones contain many built in sensors, including a GPS and 3G antenna.
2. The back camera is far superior to the built in stabilization camera of the AR.Drone
3. Electronics are self-contained, reducing drag on the UAV.

4. We are able to use the Android APIs to interface with sensors.
5. The data connection allows the phone to relay telemetry data and receive instructions from a remote server.

However, the heavy phone (129 grams) decreased the battery life of the UAV and the maximum altitude. To increase the cargo capacity of the AR.Drone, we removed the drone hull and phone casing. We experimented with increasing lift with helium balloons, as a a 30 cm balloon can lift 14 grams.

3 Results

Our system captures outdoor aerial imagery, First Person View panoramas and 3D indoor point clouds. Each intelligent quadrotor drone is under $500 per unit. It has a maximum angular resolution of under $5cm$, while the GeoEye1 satellite has an angular resolution of 41 cm; our drones are almost an order of magnitude better. [1]

3.1 Map Stitching

Raw frames from each drone streamed from the drones for preliminary processing are overlain on Google Maps without stitching to give responders imagery as quickly as possible.

Photos are stitched together into full maps with the computer vision library OpenCV[8]. Features are matched with the Scale-invariant feature transform (SIFT). The process can be slow and imperfect, although optimizations are made using the drone flight plan.

Stitched maps: Figure 16, Figure 17, Figure 18, Figure 19

3.2 Panoramic Imaging Spheres

At clusters of image key points, a drone descends to a low altitude and spins in a complete circle while records footage with the 720p front camera. Frames from this video are stitched together, creating a 360° panoramic sphere of an area (Figure 6, Figure 20 in Illustrations). **Objects as**

[1]Resolution in the context of aerial imagery is defined to be the minimum distance between two resolvable features

small as 5 cm are discernible in this panorama, an order of magnitude better than satellite imagery.

Figure 6: Full panorama from a backyard sweep

3.3 Face Detection

BRIEF: Faces are detected within generated panoramas with a two pass algorithm we designed to manage large images. **Much of the search process in disaster recovery can be automated with our system.**

Facial features are discernible in the panoramic imaging spheres generated by low altitude drones searching for survivors. In order to eliminate the time consuming search process disaster responders must manually do even with aerial imagery, faces are detected and reported to rescuers. We evaluated three existing implementations of face detection algorithms for accuracy over a set of 39 images: the Haar feature detection in OpenCV, the FindFaces Mathematica function and Chan Vese image binarization.

Figure 7: ChanVeseBinarize

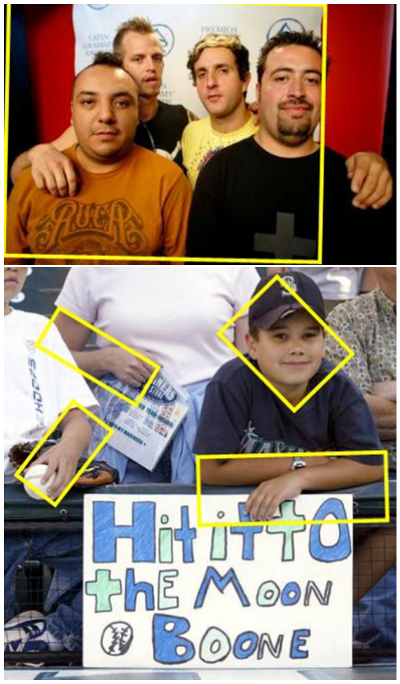
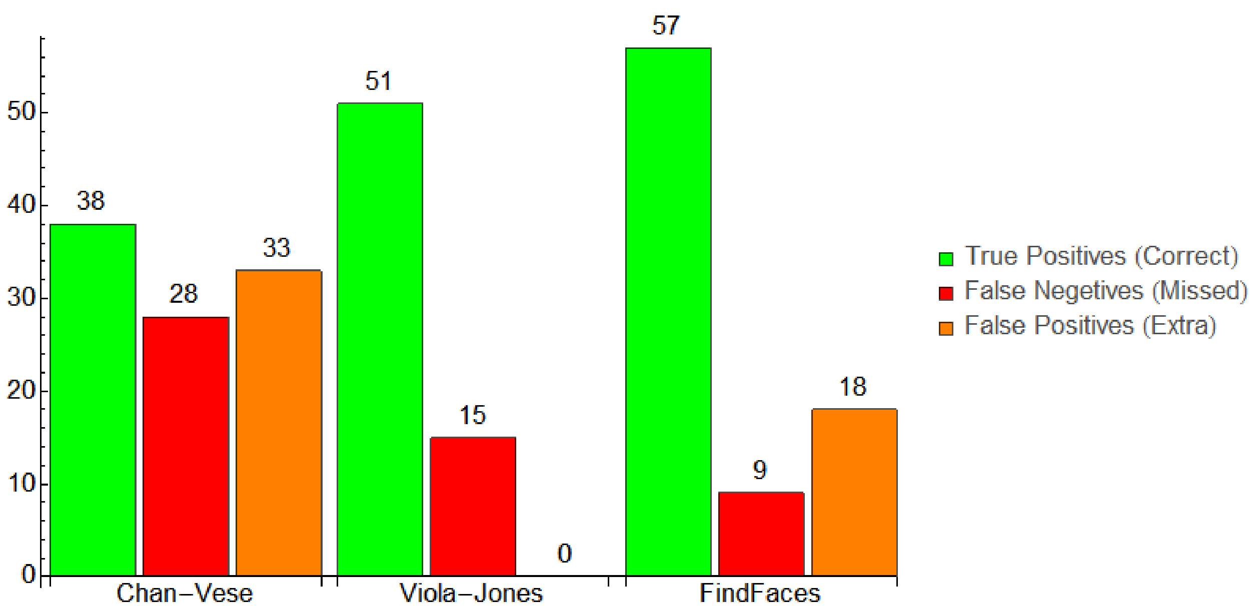


Figure 8: Haar-Feature detection

Figure 9: FindFaces

Figure 10: Face detection algorithm accuracy comparison



3.3.1 FindFaces and Viola-Jones

Large XML files are distributed with OpenCV describing pre-generated Haar features. These are created ahead of time with a machine learning algorithm where faces are manually selected in a large data set. The Haar feature detection functions in OpenCV implement the Viola-Jones object detection algorithms. Other features—trees, eyes, and animals, for example—can be detected trivially by swapping out the data set.

FindFaces and the Haar feature detection method had few false positives and negatives (extra and missed faces) (Figure 10), but perform badly on large images. The image is usually scanned region by region, which is infeasible for our panoramas.

3.3.2 Chan Vese Image Binarization

The Chan Vese algorithm segments (splits) an image, locating the boundaries of an image. In the Mathematica ChanVeseBinarize implementation [9], objects of a target color are segmented out. By targeting skin tones, specifically orange, we were able to detect exposed skin in an image. Used exclusively for face detection, the algorithm returns many false positives since all skin and orange objects are identified. However, the algorithm can process large images far faster than Haar feature detections and FindFaces. **To detect faces in large panoramas, we first narrow the possible search space to skin colored areas with ChanVeseBinarize, then apply one of the more robust algorithms to reduce false positives.**

3.4 3D Point Cloud Generation

Figure 11: Depth map created by RGBD camera.

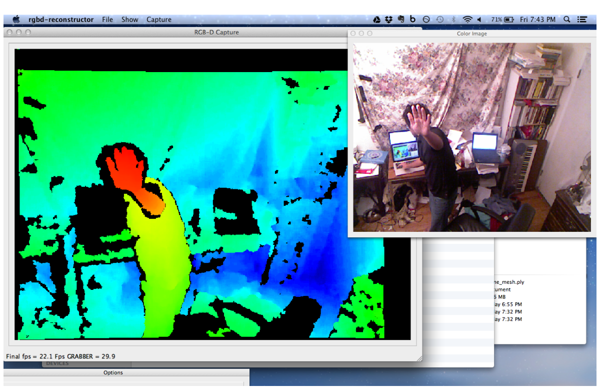


Figure 12: Generated point cloud. 3D map can be sent to doctors for remote diagnosis of victims.

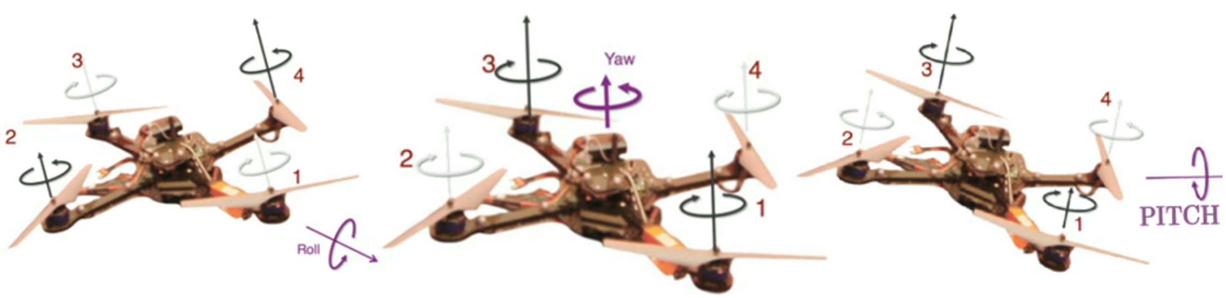
Using a stripped down Asus Xtion, a consumer depth camera originally intended for gaming, RGB-D data could be obtained from an drone indoors with limited range (Figure 11). SLAM (Simultaneous Localization and Mapping) is used to generate 3D point clouds of the interior of a building (Figure 12). **These point clouds can be sent to off-site doctors for remote diagnosis of victims. First responders do not need to enter dangerous collapsing buildings after a disaster until they are certain there is someone in need inside.**

3.5 Google Earth Map Creation

Map data with stiches superimposed upon Google Maps is delivered to first responders through KML export for portability and ease of use by First Responders. Pictures are overlaid on a satellite map and oriented and sized from drone telemetry.[2] **Operators and first responders can assess damage in regions and determine where help is needed while locating survivors.**

4 Illustrations

Figure 13: Quadrotor dynamics[10]



[2]This telemetry includes latitude/longitude, heading, bearing and altitude

Figure 14: Mapping system breakdown

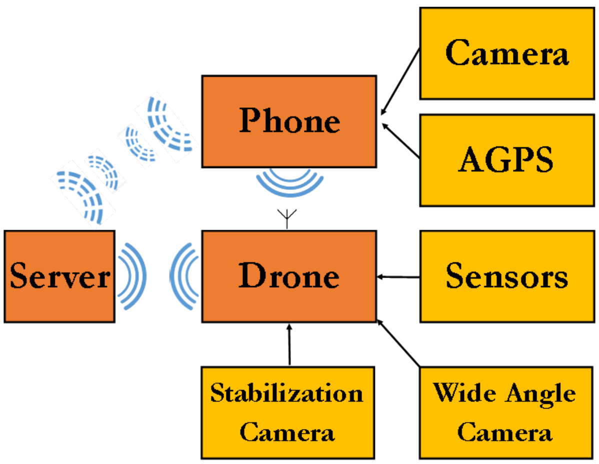


Figure 15: Arduino interfacing with drone

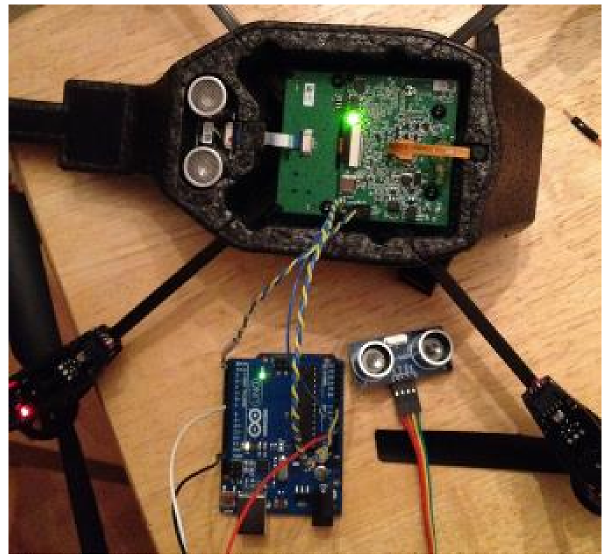


Figure 16: Stitch from night-time UAV flyover

Figure 17: Comparison of satellite and stitch quality

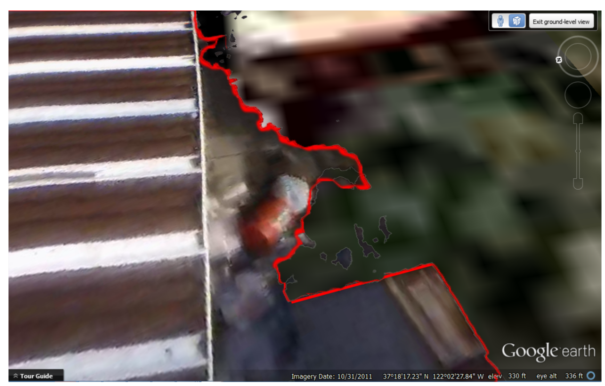


Figure 18: Stitched image of a playground

Figure 19: Map created from house flyover

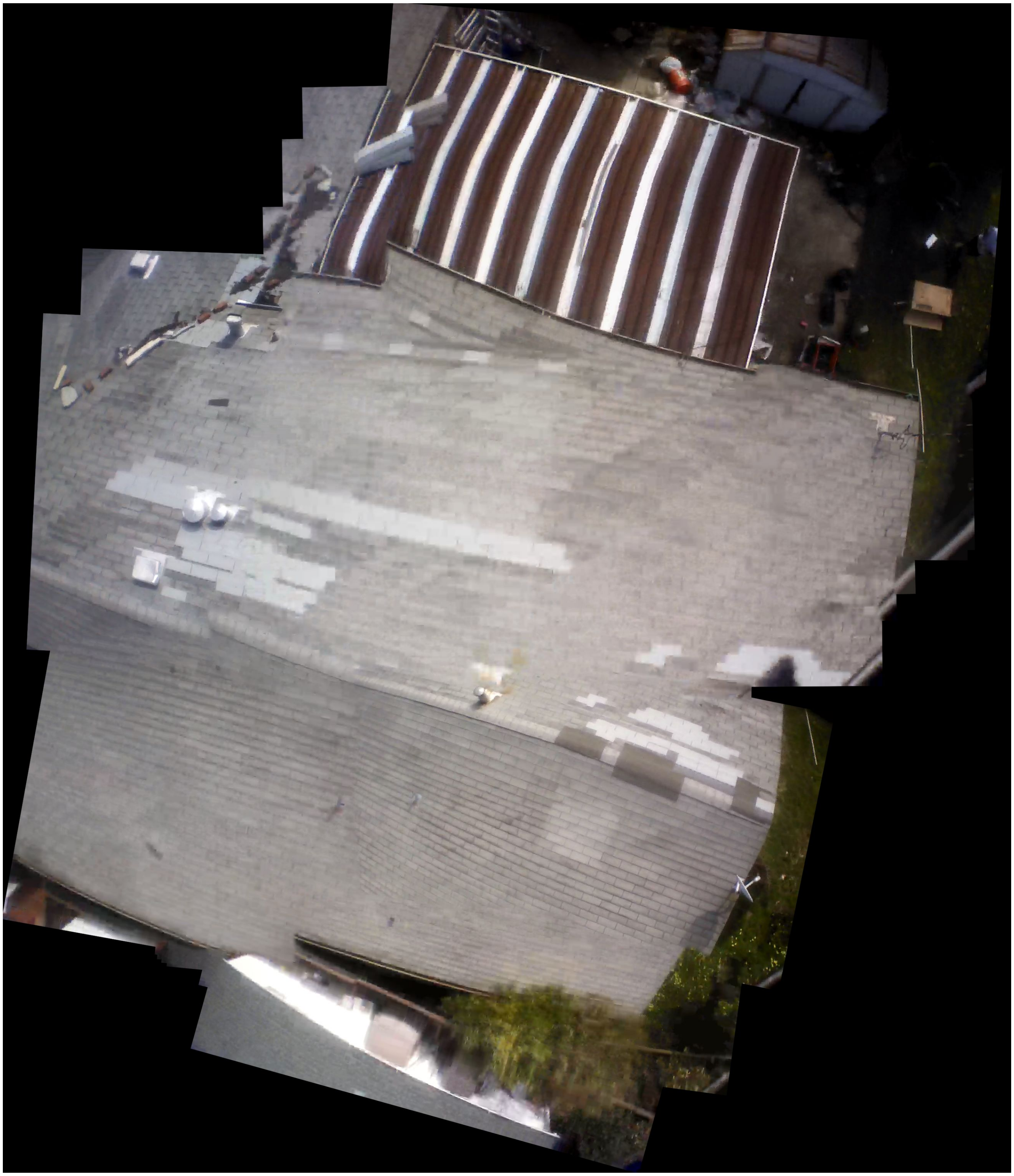


Figure 20: Panoramic stitched image from 720 p front camera above rooftops

5 Discussion

5.1 Design Criteria Analysis

Our final prototype met all of our initial design criteria, satisfying the engineering goal.

1. The UAV can withstand wind speeds up to 15 km/h without significant degradation in flight ability. Drifting does occur, however, and higher wind speeds may result in a drone becoming unpredictable. Winds past 30 km/h could result in the drone bailing due to inversion. However, given enough altitude, the drone can recover from free-fall.
2. A GPS unit, barometer, ultrasonic distance sensor and an accelerometer were used.
3. $500 total net cost for a single unit
4. We included a 720p front camera for generating panoramas, 5 megapixel bottom facing phone camera for mapping, 240p stabilization camera. In panoramas, objects as small as 5 cm can be resolved - this is an **order of magnitude better than satellite imagery** and allows for victim detection.

5.2 Impact and Applications

5.2.1 Disaster Response

Aerial maps allow for fast damage assessment. Imagery is automatically analyzed for faces and keypoints, then stitched and superimposed onto maps. Load is taken off emergency and recovery workers, allowing their efforts to be focused on investigating the detected interest areas. Up to date maps of fire, flood, earthquake and hurricane damaged areas are created where the delay from satellites is intolerable.

5.2.2 Search and Rescue

Since the PC server handles drone flight, operators only need to select a search area. This allows rapid situation assessment without having to wait for trained aircraft or helicopter pilots to arrive. Fast response is critical in search and rescue operations where survival chances decline very quickly

5.2.3 Indoor Mapping

Search and rescue teams can 3-dimensionally map the interior of damaged buildings to locate survivors without self-endangerment. The maneuverability of quadrotors allows navigation through doors, windows and skylights.

5.3 Comparison to Alternate Solutions

SATELLITES: The final system produces images with up to 10 times higher angular resolution than the best available satellite imagery. A network of our UAVs can be deployed over a city immediately after a disaster, whereas imagery from satellites would not be available for several days.

PILOTED AIRCRAFT: A full scale helicopter can cover much more ground than an individual quadcopter, but must fly at a high altitude, limiting the resolution of imagery. The coverage limitation was addressed by developing a network of multiple drones that map a city in pieces and progressively enhance the resolution of devastated areas. Additionally, none of the alternate solutions address the problem of indoor mapping.

OTHER UAVs: Each of our quadrotor systems is under $\$500$, far more inexpensive than existing UAVs that have been used for search and rescue. Many alternate UAVs are piloted and simply return a video feed. This requires at least one operator per drone, when manpower should be focused on rescue.

6 Conclusions and Future Work

We effectively produce up to date, high resolution maps and models that assist with fast damage assessment in disaster response, search and rescue, and indoor survivor search. Our system has an order of magnitude better angular resolution than current satellite imaging, and each drone is under $500, compared to current UAVs ranging from tens to hundreds of thousands of dollars.

6.1 How we met our Project Goals

After developing 3 main hardware prototypes, our final drone uses a self contained image and telemetry collection unit, while initially we used the Arduino microprocessor. Our system 1) captures low resolution imagery with a high altitude drone to 2) automatically identify damaged areas where 3) low altitude drones capture very high resolution imagery. Data is 4) presented on existing aerial mapping tools used by first responders.

Within dangerous buildings, quadrotors with 3D cameras capture full 3D maps of building interiors. Bodies are detected and the models are sent to doctors for remote diagnosis. This technology saves the lives of first responders as they do not need to search inside collapsing buildings.

6.2 Further Research

In the future, we will investigate alternate UAV technologies to allow for more carry-weight. In turn, we could mount higher quality imaging equipment and additional sensors. After looking into other rotor based systems, we would develop an imaging platform based on a fixed-wing UAV. Fixed wing planes can cover large areas in one flight, although we would not be able to use the same system to map the interiors of buildings.

As we develop the project, we may develop an networking solution between drones without a centralized PC server. Tasks and computation would be divided between drones with peer to peer communication. This would allow for massive scale deployments of drones across a city.

7 References

7.1 References

[1] N. Halftermeyerc. Parrot ar.drone 2.0 flying. [Online]. Available: http://en.wikipedia.org/wiki/File:Parrot_AR.Drone_2.JPG

[2] M. Martinez. Police, citizens, technology lead to boston bombing manhunt's success. CNN. [Online]. Available: http://www.cnn.com/2013/04/20/us/boston-capture

[3] Google. Google Earth release notes. [Online]. Available: https://support.google.com/earth/answer/40901

[4] C. Waite. Aeryon labs drone bid for alameda county police. [Online]. Available: http://www.wired.com/images_blogs/threatlevel/2012/12/Aeryon-bid.pdf.pdf

[5] S. Shankland. Google to buy geoeye satellite imagery. CNET. [Online]. Available: http://news.cnet.com/8301-1023_3-10028842-93.html?part=rss&subj=news&tag=2547-1023_3-0-5

[6] A. Campoy. The law's new eye in the sky: Police departments' use of drones is raising concerns over privacy and safety. The Wall Street Journal. [Online]. Available: http://online.wsj.com/article/SB10001424052970204319004577088891361782010.html

[7] Ar.drone 2.0 technical specifications. Parrot. [Online]. Available: http://ardrone2.parrot.com/ardrone-2/specifications/

[8] Opencv. WillowGarage. [Online]. Available: http://opencv.org/

[9] ChanVeseBinarize. Wolfram. [Online]. Available: http://reference.wolfram.com/mathematica/ref/ChanVeseBinarize.html

[10] V. Kumar. (2012, Mar.) Robots that fly and cooperate. TED. [Online]. Available: http://www.ted.com/talks/vijay_kumar_robots_that_fly_and_cooperate.htm

[11] H. Dersch. Panorama tools: Open source software for immersive imaging. [Online]. Available: http://webuser.fh-furtwangen.de/~dersch/IVRPA.pdf

[12] J. D. Bell. Getting started with python and opencv. NeuroForge. [Online]. Available: http://www.neuroforge.co.uk/index.php/getting-started-with-python-a-opencv

[13] L. G. Thuy. Setup opencv for python in windows. [Online]. Available: http://luugiathuy.com/2011/02/setup-opencv-for-python/

[14] T. Krajnik. Ardrone quadcopter in robotics research. [Online]. Available: http://labe.felk.cvut.cz/~tkrajnik/ardrone/

[15] Hurricane sandy: The aftermap. Esri. [Online]. Available: http://www.esri.com/services/disaster-response/hurricanes/hurricane-sandy-the-aftermap

[16] A. Méchaly. One flew over the cornfield. Alcatel Lucent. [Online]. Available: http://www2.alcatel-lucent.com/blogs/corporate/2012/10/one-flew-over-the-cornfield/

[17] Dkroetsch, "Aeryon Scout Micro with gyro stabilized camera payload," Wikipedia. [Online]. Available: http://commons.wikimedia.org/wiki/File:Aeryon_Scout_With_Camera.jpg