

# 3D Model Generation of Large Objects Using Autonomous Quadcopters

Edward Kelley  
Advisor: Szymon Rusinkiewicz

## 1. Introduction

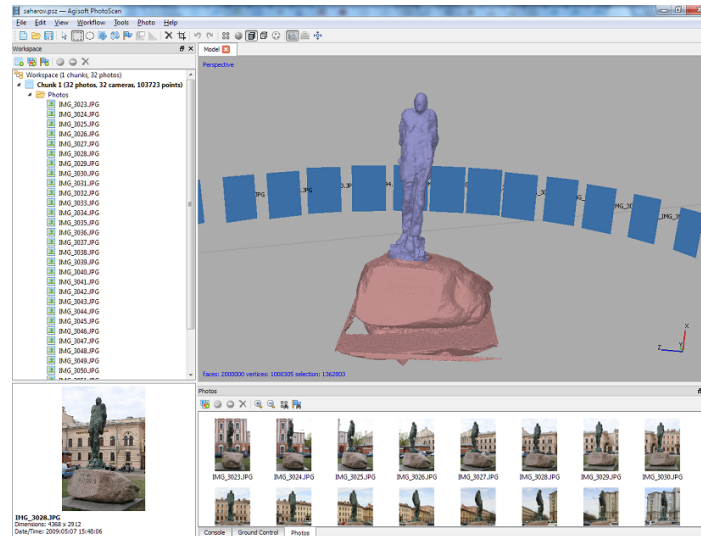
### 1.1. Background

For applications ranging from developing video games to preserving archaeological artifacts, capturing 3D models of real-world objects has become a very important task. Several different techniques for acquiring models have been developed, such as the use of laser scanners and multi-view stereo.

**1.1.1. Laser Scanners** Laser rangefinder technology is the “gold standard” of 3D model acquisition in terms of quality. Modern scanners can produce sub millimeter accuracy, which make them a great choice for detailed digitization of statues. Combined with high-resolution photograph texture-mapping, very few techniques can match the precision and quality of these scans. The Digital Michelangelo Project showed the power and precision of laser scanners by scanning several different statues, including Michelangelo’s David, to 1/4mm accuracy.[9]



Figure 1: An example of a laser scanner setup used by the Digital Michelangelo Project [9].



**Figure 2: A 3D model of a statue generated by Agisoft Photoscan. Notice the derived camera planes encompassing the statue [1].**

However, laser scanners do have several drawbacks. The equipment is extremely expensive, bulky, and fragile. The Michelangelo Project had to transport over 4 tons of equipment to Italy in order to produce their scans. Additionally, laser scans involve immense setup and can take many hours. The scan of David took over a thousand man-hours to scan and even more than that in post processing [9].

**1.1.2. Multi-View Stereo** Multi-view stereo uses a collection of 2D images to reconstruct a 3D object model. By viewing a single object from hundreds of different camera positions, a 3D model can be generated. Although this technique originally required precisely known camera coordinates, recent algorithms can produce a 3D model from an unordered collection of images with unknown camera positions, assuming that there is sufficient coverage. Existing software packages such as Bundler and Agisoft Photoscan can produce high-quality 3D reconstructions using these unordered image collections. [2][1]

The ability to use a collection of images without precise camera position information means that these 3D objects can be modeled substantially faster than with a laser scanner. With a smaller object, it is a relatively simple process to take pictures of the object from many different angles. However, for a larger object, such as a statue or building, the problem of gathering imagery becomes substantially more difficult.

## 1.2. Problem Definition

We look to create a system to capture imagery of large 3D objects for use in multi-view stereo software. This system has several requirements.

### 1. Low Cost

The system should be substantially cheaper than laser scanning.

**2. Easy to Use**

This system should be able to be deployed by users with minimal training. Additionally, the hardware should be off-the-shelf and easily accessible.

**3. Complete Coverage**

The system must be able to capture images from a wide variety of positions, completely covering every part of the target object.

**4. High Quality Imagery**

The system must produce sufficiently high resolution, non-blurry images for use in multi-view stereo software.

### **1.3. Proposed Solution**

We propose using low-cost autonomous quadcopters to gather imagery needed for use in multi-view stereo software. By flying a quadcopter with a mounted camera around the target object, we can quickly and thoroughly capture images of the target from a wide variety of positions. Using quadcopters has many advantages.

1. Quadcopters can capture images from positions unreachable by ground-based cameras.
2. By methodically flying around the target object at different altitudes, we can guarantee complete coverage of the target object.
3. The imagery can be captured very quickly, on the order of a few minutes.
4. Quadcopters are small, portable, and easily deployable.

### **1.4. Hardware**

For this project, we are using the Parrot AR.Drone 2.0. Although originally designed as a high-tech toy, many have found the AR.Drone to be a useful platform for research. At only around \$300, the AR.Drone is substantially less expensive than most quadcopters which typically cost a couple thousand dollars. However, despite the low cost, the AR.Drone boasts an impressive array of sensors and technology.

The drone has two cameras, one forward-facing 720p wide-angle, and one downward-facing QVGA camera. The drone has an on-board processor to automatically stabilize itself using its gyroscope, accelerometer, and magnetometer readings. Additionally, the drone has both ultrasound and pressure sensors for altitude measurement.

### **1.5. Related Work**

The past decade has seen a huge increase in the use of quadcopters for a variety of applications. With the improvement of stabilization software, quadcopters have seen a rise in popularity as a



**Figure 3: The Parrot AR.Drone 2.0 with Indoor Hull**

stable, cheap, and highly maneuverable aerial platform.

Although a relatively new field, several research groups have studied the use of quadcopters in 3D model construction. Irschara et al. created a system to generate 3D models using images taken from UAVs. While a quadcopter was used for gathering imagery, the quadcopter was manually controlled and the main focus of their work was photogrammetry-based model creation [7]. Steffen et al. studied surface reconstruction using aerial photography captured by UAVs [11].

Most relevant to our work, Engel et al. published multiple papers on the camera-based navigation and localization of the AR.Drone. While they were able to achieve very accurate navigation, their work relies on the drone facing a mostly planar surface during the entire flight, a constraint that is not possible in our application.

## **2. Project Status**

### **2.1. Quadcopter Interface**

After doing substantial research, we decided on using “ardrone\_autonomy” a Robot Operating System (ROS) driver for the AR.Drone. This package allows us to easily subscribe to the navdata from the drone as well as publish commands. ROS provides a nice framework for organizing different elements of our system as well as integration with libraries such as OpenCV.

### **2.2. Software**

Our first step was to build a ROS package that would allow us to issue basic commands to the drone and receive video. We built a graphical user interface so that we could easily control the drone

manually. We then worked on using the OpenCV library with the video feed. Currently, we can track detected features between frames, which will be important for estimating object coverage. We have also started to work on the problem of localization. We built a particle filter that will propagate a position estimate of the drone based on the velocity readings from the drone.

### **2.3. Next Steps**

After testing the drone with basic commands, we realized that localization will be a major issue. Using only the dead reckoning velocity measurements of the drone, the position estimate becomes very inaccurate after only a few instructions. Therefore, we are going to look into using visual tags on the ground along the bounding box of the flight path. We can use the computed relative position to these tags to greatly increase the accuracy of our position estimate. Additionally, these may help us compensate for some of the wind drift we have been experiencing. After producing a more reliable position estimate, we can work on motion planning and coverage estimation.

### **2.4. Timeline**

**2/18/13** Determine relative position to visual tags.

**2/25/13** Localization correction using visual tags.

**3/11/13** Fly programmed pattern around a statue.

**3/25/13** Create voxel representation of explored/unexplored space.

**4/1/13** Test system on different objects. Start writing report.

**4/22/13** Full rough draft of final report.

**5/6/13** Final draft turned in.

## References

- [1] “Agisoft photoscan.” [Online]. Available: <http://www.agisoft.ru/products/photoscan>
- [2] “Bundler: Structure from motion (sfm) for unordered image collections.” [Online]. Available: <http://phototour.cs.washington.edu/bundler/>
- [3] P.-J. Bristeau *et al.*, “The navigation and control technology inside the ar.drone micro uav,” 2011.
- [4] N. Dijkshoorn, “Simultaneous localization and mapping with the ar.drone,” 2012.
- [5] J. Engel, J. Sturm, and D. Cremers, “Accurate figure flying with a quadcopter using onboard visual and inertial sensing,” *IMU*, vol. 320, p. 240.
- [6] J. Engel, J. Sturm, and D. Cremers, “Camera-based navigation of a low-cost quadcopter,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, oct. 2012, pp. 2815–2821.
- [7] A. Irschara *et al.*, “Towards fully automatic photogrammetric reconstruction using digital images taken from uavs,” in *Proceedings of the ISPRS TC VII Symposium—100 Years ISPRS*, 2010.
- [8] K. Leung, C. Clark, and J. Huissoon, “Localization in urban environments by matching ground level video images with an aerial image,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, may 2008, pp. 551–556.
- [9] M. Levoy *et al.*, “The digital michelangelo project: 3d scanning of large statues,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH ’00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 131–144. Available: <http://dx.doi.org/10.1145/344779.344849>
- [10] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, “Real-time 3d model acquisition,” 2002.
- [11] R. Steffen and W. Förstner, “On visual real time mapping for unmanned aerial vehicles,” in *21st Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, 2008, pp. 57–62.
- [12] T. Yap *et al.*, “A particle filter for monocular vision-aided odometry.”