A PRELIMINARY STUDY ON UPDATING HIGH DEFINITION MAPS: DETECTING AND POSITIONING A TRAFFIC CONE BY USING A STEREO CAMERA

M. L. R. Lagahit *, Y. H. Tseng

Dept. of Geomatics, National Cheng Kung University, Tainan, Taiwan – (miguellagahit, tseng)@prs.geomatics.ncku.edu.tw

Commission IV

KEYWORDS: YOLO, Stereo-photogrammetry, Object Detection, High-Definition Maps, Autonomous Vehicles

ABSTRACT:

The concept of Autonomous Vehicles (AV) or self-driving cars has been increasingly popular these past few years. As such, research and development of AVs have also escalated around the world. One of those researches is about High-Definition (HD) maps. HD Maps are basically very detailed maps that provide all the geometric and semantic information on the road, which helps the AV in positioning itself on the lanes as well as mapping objects and markings on the road. This research will focus on the early stages of updating said HD maps. The methodology mainly consists of (1) running YOLOv3, a real-time object detection system, on a photo taken from a stereo camera to detect the object of interest, in this case a traffic cone, (2) applying the theories of stereo-photogrammetry to determine the 3D coordinates of the traffic cone, and (3) executing all of it at the same time on a Python-based platform. Results have shown centimeter-level accuracy in terms of obtained distance and height of the detected traffic cone from the camera setup. In future works, observed coordinates can be uploaded to a database and then connected to an application for real-time data storage/management and interactive visualization.

1. INTRODUCTION

1.1 Autonomous Vehicles

Figure 1. Google’s Self Driving Car
(taken from reuters.com)

In the year 2016 almost 2000 research works related to Autonomous Vehicles (AV) or self-driving cars have been published (Brummelen, 2018). The sheer amount of publications done just proves that the concept of AVs is one of the most popular uprising topics in the field of science and engineering.

But, how can you say that a vehicle is autonomous? Well, we can start by taking a look at the entire process of “autonomous” navigation. It can be simply categorized into five components: (1) perception – where the AV detects objects and features in its surrounding environment, (2) localization and mapping – where the AV positions itself on the road and relatively positions everything it has detected to itself, (3) path planning – where the AV determines what route/s to take, (4) decision making - where the AV decides on how it will interact with everything else on the road, and (5) vehicle control – where the AV internal program synchs with its corresponding mechanical parts (Brummelen, 2018).

This research will be focusing on the components of perception, and localization and mapping, in terms of their relations to High-Definition (HD) Maps.

1.2 High-Definition Maps

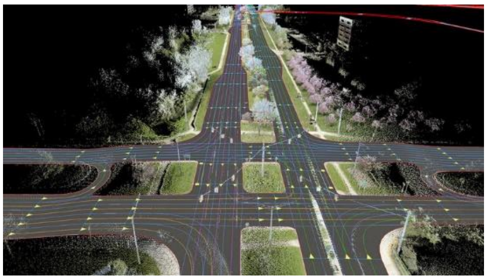


Figure 2. HD Map
(taken from geospatialworld.net)

HD Maps are 3D, centimeter-level precision, maps produced from survey grade Mobile Mapping Units (MMU) that helps AVs better localize itself and features nearby to make its navigation on the road easier. It is composed of 4 layers: (1) the geometric layer – which contains raw and processed sensor data, (2) the semantic layer – which contains information that can be found on objects and markings, (3) the map prior layer – which contains recorded human behavior patterns in an area, and (4) the real-time layer – which contains real-time traffic data (Vardhan, 2017). Using combinations of those layers makes AVs wiser in times of critical conditions such as: driving on roads nearby

* Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-4-W19-271-2019 | © Authors 2019. CC BY 4.0 License.

cliffs, on heavily crowded areas, and on bad weather conditions like heavy rain.

Nonetheless, the technology of HD Maps is still relatively new and is still in development. There are a lot of things that are still in need of improvement such as updating, making it adapt to real-time changes happening in the AV environment (Brummelen, 2018). Just imagine if you were to deploy MMUs on a daily basis, not only would it be costly but also inefficient since changes might happen in less than a span of a day. This makes integrating updating HD Maps in the AV system much more important so that an AV will not only use an HD Map but improve it as well.

This research will be working on that problem by proposing a preliminary updating framework of using a stereo camera to detect and position a traffic cone.

1.3 You Only Look Once (YOLO)

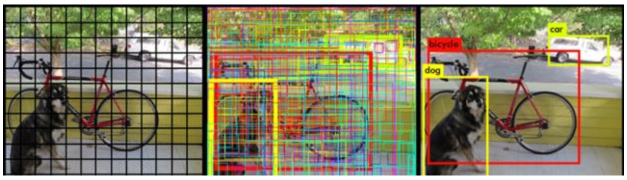


Figure 3. YOLO's Object Detection Process
(taken from pjreddie.com)

For the detection part of the proposed framework, You Only Look Once (YOLO) will be used. YOLO is a real-time object detection system developed by Joseph Redmon and Ali Farhadi. Now on its 3rd version (YOLOv3), it uses a total of 53 convolutional layers and boasts of speeds 100x faster than Fast R-CNN.

In a study done by Xiang Zhang and his/her colleagues, they used and compared YOLOv3 with other object detection systems in detecting lane markings using the KITTI and Caltech datasets. Their results have shown that YOLOv3 has a mean average precision of 88.39% and 89.32% with speeds of 25.2 ms and 24.7 ms, respectively, which is a lot better than the RCNN approach which only resulted in 79.26% and 81.75% mean average precision with speeds of 29.3 ms and 25.6 ms, respectively. As such, it can be said that YOLOv3 is accurate and fast enough to compete with rivaling systems used for AV purposes.

1.4 Objectives and Limitations

The main objective of this research is to be able to detect and position a single stationary traffic cone using a stereo camera as the only sensor. Data acquisition will be done in a single set-up station, meaning the stereo camera will also be stationary. During the time the pictures were taken on fairly clouded weather in the afternoon.

This research will not factor in results from varying lighting conditions, target occlusion, feature tracking, and positioning multiple objects. Both a moving target and a moving sensor will also not be included in the experiment.

2. METHODOLOGY

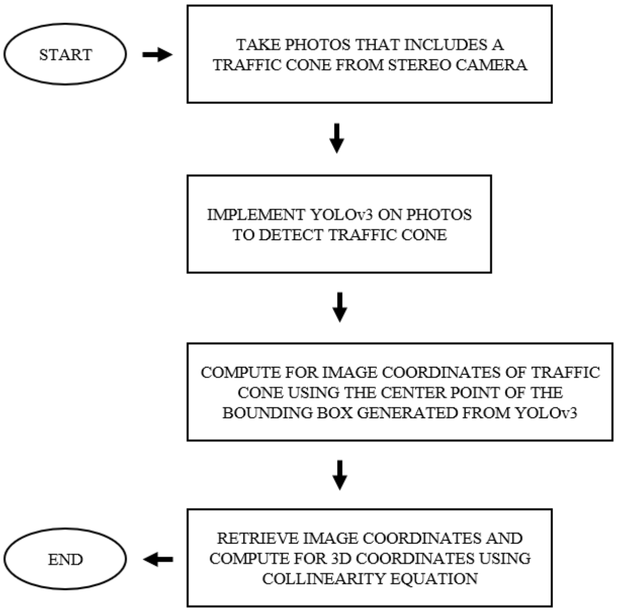

START
TAKE PHOTOS THAT INCLUDES A TRAFFIC CONE FROM STEREO CAMERA
IMPLEMENT YOLOv3 ON PHOTOS TO DETECT TRAFFIC CONE
COMPUTE FOR IMAGE COORDINATES OF TRAFFIC CONE USING THE CENTER POINT OF THE BOUNDING BOX GENERATED FROM YOLOv3
RETRIEVE IMAGE COORDINATES AND COMPUTE FOR 3D COORDINATES USING COLLINEARITY EQUATION
END

Figure 4. General Methodology of the Research

Figure 4 shows the custom-made stereo camera. It is composed of two Sony α6000 cameras fixed on a piece of wood 27 centimeters apart. Then, the stereo camera's interior and exterior (relative to the right camera) orientation parameters are computed using MATLAB R2019a's stereo camera calibrator. For this research, lens distortion correction is not included in the process to simulate positioning using raw images without any sort of modification.

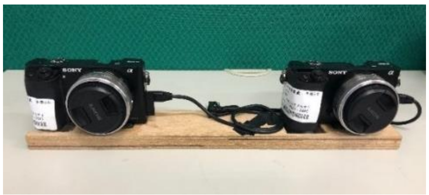


Figure 5. Stereo Camera

A pair of photos of scenery containing a traffic cone is then taken using the stereo camera, as shown in Figure 5. Its distance with respect to the camera is also measured using a steel tape to manually check the generated results of the proposed methodology.

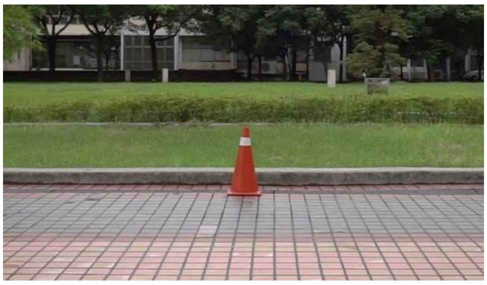


Figure 6. Cropped Photo Containing a Traffic Cone
(Taken from the Left Camera)

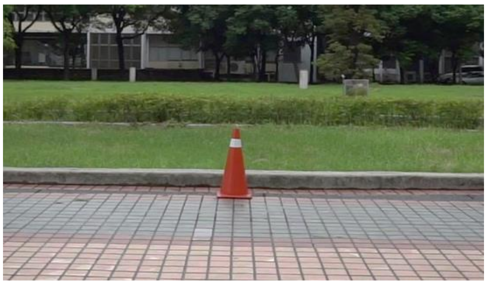


Figure 7. Cropped Photo Containing a Traffic Cone (Taken from the Right Camera)

Using a computer having specifications of an Intel i7 processor, 16 GB of RAM, and an NVIDIA GeForce GTX 1060 graphics card a custom trained YOLOv3 python script is executed on the pair of photos to detect the traffic cone. Figure 8 and 9 shows the result of the YOLOv3 implementation on the photo in Figure 6 and 7, respectively. It is shown that YOLOv3 encloses the object of interest, in this case, a traffic cone, in a rectangular bounding box. The center of this bounding box will then later be used to represent the traffic cone's position.

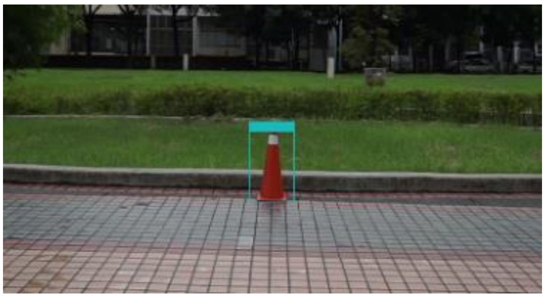


Figure 8. Cropped YOLOv3 Result (Left Image)

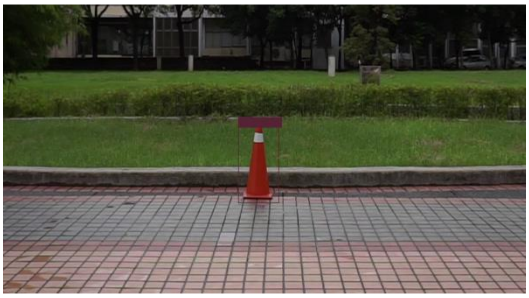


Figure 9. Cropped YOLOv3 Result (Right Image)

Using the generated interior and exterior parameters from the camera calibration and the image coordinates from YOLOv3, the relative 3D position of the traffic cone is calculated using a Python script of a least-square iterated collinearity equation shown in Equation 1 (Förstner & Wrobel, 2016). Delta (Δ) is computed using the previously generated parameters along with an initial assumption of the values of the traffic cone's position. The process is done by adding Delta to the initially assumed position and using the result as the new value of X, Y, and Z. This process will be repeated until the sum of the squares of the values of Δ is greater than 10^{-10}.

$$\Delta = (B^T B)^{-1}(BF)$$

$$F = \begin{bmatrix} x - f\dfrac{N_x}{N_z} \\ y - f\dfrac{N_y}{N_z} \\ \dots \end{bmatrix}$$

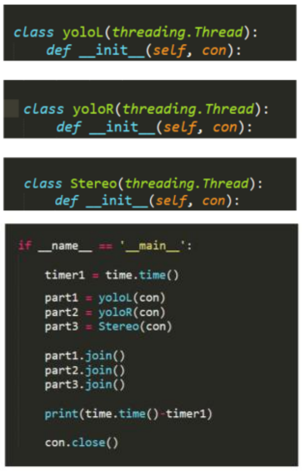
$$B = \begin{bmatrix} \dfrac{\partial x}{\partial X} = -f\dfrac{r_{11} \cdot N_z - r_{13} \cdot N_x}{(N_z)^2} & \dfrac{\partial x}{\partial Y} = -f\dfrac{r_{21} \cdot N_z - r_{23} \cdot N_x}{(N_z)^2} & \dfrac{\partial x}{\partial Z} = -f\dfrac{r_{31} \cdot N_z - r_{33} \cdot N_x}{(N_z)^2} \\ \dfrac{\partial y}{\partial X} = -f\dfrac{r_{12} \cdot N_z - r_{13} \cdot N_y}{(N_z)^2} & \dfrac{\partial y}{\partial Y} = -f\dfrac{r_{22} \cdot N_z - r_{23} \cdot N_y}{(N_z)^2} & \dfrac{\partial y}{\partial Z} = -f\dfrac{r_{32} \cdot N_z - r_{33} \cdot N_y}{(N_z)^2} \\ \dots & \dots & \dots \end{bmatrix}$$

$$N_x = r_{11} \cdot (X - X_0) + r_{21} \cdot (Y - Y_0) + r_{31} \cdot (Z - Z_0)$$
$$N_y = r_{11} \cdot (X - X_0) + r_{21} \cdot (Y - Y_0) + r_{31} \cdot (Z - Z_0)$$
$$N_z = r_{11} \cdot (X - X_0) + r_{21} \cdot (Y - Y_0) + r_{31} \cdot (Z - Z_0) \qquad (1)$$

Where,

$X, Y, Z =$ *object coordinates of traffic cone*
$x, y =$ *image coordinates of traffic cone*
$f =$ *focal length (from camera calibration)*
$X_0, Y_0, Z_0 =$ *exterior orientation of stereo camera (from camera calibration)*
$r_{nm} =$ *value of row n and column m of the rotational matrix (from camera calibration)*

The whole process was running at the same time to simulate a real-time workflow, similar to what an AV would process. It is executed using a Python script shown in Figure 10. It contains 3 classes: 2 classes calculate the traffic cone's image coordinates using YOLOv3 on each camera and 1 class to calculate the 3D position. Since everything is running at the same time, the third class is made to continuously run to await the other 2 classes resulting information.



```
class yoloL(threading.Thread):
    def __init__(self, con):
```

```
class yoloR(threading.Thread):
    def __init__(self, con):
```

```
class Stereo(threading.Thread):
    def __init__(self, con):
```

```
if __name__ == '__main__':

    timer1 = time.time()

    part1 = yoloL(con)
    part2 = yoloR(con)
    part3 = Stereo(con)

    part1.join()
    part2.join()
    part3.join()

    print(time.time()-timer1)

    con.close()
```

Figure 10. Parts of the Python Script of the Proposed Methodology

3. RESULTS AND DISCUSSION

The results of the camera calibration done in MATLAB are shown in Tables 1 and 2. The generated focal lengths only have differences of about 0.1 mm for both the cameras from Sony's indicated 16 mm focal length for the model α6000. The generated distance between the cameras also only has about a 0.44 cm difference from the manually measured value.

Table 1. Interior Orientation Parameters in Millimeters

	Value	Estimated Error
Left Camera		
Focal Length	15.9004	0.0151
Principal Point (X)	11.5591	0.0187
Principal Point (Y)	7.6025	0.0161
Right Camera		
Focal Length	15.8703	0.0152
Principal Point (X)	11.4181	0.0200
Principal Point (Y)	7.3865	0.0165

Table 2. Exterior Orientation Parameters
(Relative to the Right Camera)

	Value	Estimated Error
X	274.4578	0.1296
Y	-3.0607	0.1064
Z	-1.9746	0.5825
ω	-0.2224	0.0648
ϕ	1.7567	0.0842
κ	0.5294	0.0077

The proposed methodology has been able to provide the traffic cone's position with a difference of 8.1 cm for its distance from the camera setup and 1.4 cm for its height, as shown in Table 3, which is near the 5cm accuracy needed for HD Maps (Vardhan, 2017). The centimeter level of difference might have come from the combined resulting errors of the steel tape, the stereo camera's lens distortion, and human error (in terms of where the tapes initial and end points are placed in measuring the distance). However,

Table 3. The Computed Position of the Traffic Cone in Meters

	Steel Tape	Collinearity Equation	Difference
Ground Distance from Camera Setup to Traffic Cone	8.530	8.449	0.081
Height of Traffic Cone's Center	0.350	0.336	0.014

Finally, it took less than 20 ms for the whole process to complete, making it a fairly fast process of computing an object 3D position. It is nearly comparable to the 25 ms results of Xiang Zhang and his/her colleagues, considering that they detected multiple features.

4. CONCLUSION, RECOMMENDATIONS AND FUTURE WORK

The research has successfully detected and positioned a traffic cone using a stereo camera with below 10 cm positioning accuracy both horizontally and vertically in a span of less than 20 ms. YOLOv3 and Python have also shown that they are viable connecting platforms for image object detection and positioning, specifically for HD map purposes.

It is recommended that: (1) distance between the camera setup and the traffic cone be measured in more accurate ways such as using a laser rangefinder; (2) lens distortion corrections be applied to see if results have a significant change in terms of positioning accuracy; and (3) images should be taken on different sides and angles.

In future works: (1) the measured position of the traffic cone can be uploaded/updated to a database for better management, (2) LIDAR data be incorporated for obtaining the horizontal and vertical position, and (3) connect it to an application for real-time interactive visualization and analysis.

REFERENCES

Brummelen, J.V. et al., 2018. Autonomous vehicle perception: The technology of today and tomorrow. Elsevier: Transportation Research Part C 89, pp. 384-406.

Cruise Automation Team, 2018. WORLDVIEW, Retrieved August 13, 2019, from https://webviz.io/worldview/#/docs/guides/quick-start.

Förstner, W. & Wrobel, B., 2016. Photogrammetric Computer Vision: Statistics, Geometry, Orientation, and Reconstruction. Springer, pp. 547-615.

Redmon, J. & Farhadi, A., 2018. YOLOv3: An Incremental Improvement. arXiv.

Vardhan, H., 2017. HD Maps: New age maps powering autonomous vehicles, Retrieved August 13, 2019, from https://www.geospatialworld.net/article/hd-maps-autonomous-vehicles/.

Zhang X. et al., 2018. A Fast learning Method for Accurate and Robust Lane Detection Using Two-Stage Feature Extraction with YOLO v3. Sensors 2018 18.