

USING AFFORDABLE DATA CAPTURING DEVICES FOR AUTOMATIC 3D CITY MODELLING

B. ALIZADEHASHRAFI^a, A. Abdul-Rahman^{b,*}

^aTabriz Islamic Art University, Faculty of Multimedia, Tabriz, Iran, b.alizadehashrafi@tabriziau.ac.ir

^b3D GIS Research Lab, Dept. of Geoinformation, Faculty of Geoinformation & Real Estate,
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia. alias@utm.my

KEY WORDS: Dynamic Pulse Function, Automatic 3D modelling, AR Drone, AgiSoft Photoscan

ABSTRACT:

In this research project, many movies from UTM Kolej 9, Skudai, Johor Bahru (See Figure 1) were taken by AR. Drone 2. Since the AR drone 2.0 has liquid lens, while flying there were significant distortions and deformations on the converted pictures of the movies. Passive remote sensing (RS) applications based on image matching and Epipolar lines such as Agisoft PhotoScan have been tested to create the point clouds and mesh along with 3D models and textures. As the result was not acceptable (See Figure 2), the previous Dynamic Pulse Function based on Ruby programming language were enhanced and utilized to create the 3D models automatically in LoD3. The accuracy of the final 3D model is almost 10 to 20 cm. After rectification and parallel projection of the photos based on some tie points and targets, all the parameters were measured and utilized as an input to the system to create the 3D model automatically in LoD3 in a very high accuracy.



Figure 1. Kolej 9, UTM, Skudai (The test bed)

*B. Alizadehashrafi

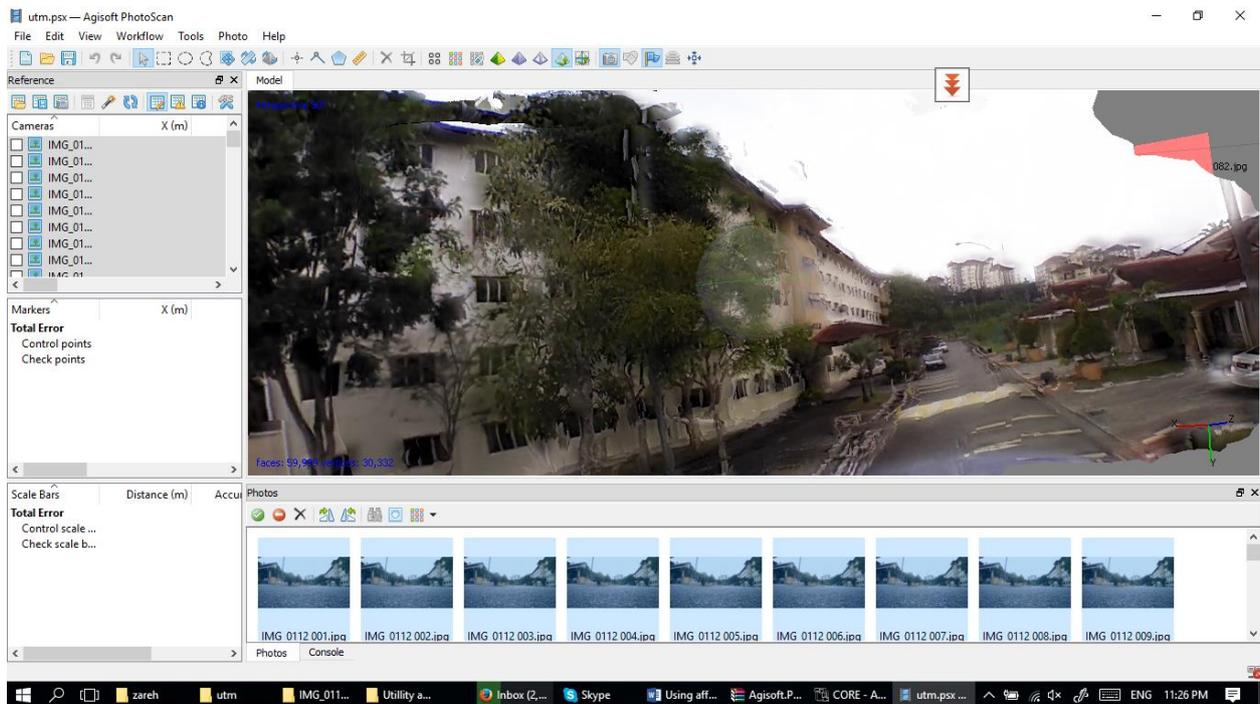


Figure 2. The result of AgiSoft PhotoScan software using 500 images extracted from the move made by Ar. Drone 2 in PSX AgiSoft format.

1. INTRODUCTION

Three-dimensional modelling is the process of developing a mathematical representation of any three-dimensional surface of an object via specialized software or programming and modeling languages such as VRML and X3D. The final product called a 3D model that used for different purposes in different formats. It can be displayed as a two-dimensional image with a high Level of Realism (LoR) through a process called 3D rendering or used in a computer simulation or Unity game engine or semantic geospatial database. The model can also be physically created using 3D printing devices which is available for many historical and heritage landmarks in old countries. The 3D Models also can be created automatically or via photo matching algorithms such as Agisoft PhotoScan (AgiSoft, 2014) and Autodesk Recap 360 (Straub and Kerlin, 2016) or programming languages. The manual modeling process of preparing geometric data for 3D computer graphics is similar to plastic arts such as sculpting. 3D modeling software is a class of 3D computer graphics software used to produce 3D models. Individual programs of this class are called modeling applications or modelers such as SketchUp, 3Ds Max, Cinema4D, Ulead, blender and so on. In this project an Apple AR Drone 2 with the price of roughly 1300 Malaysian Ringgit have been used for the purpose of 3D modeling. The most important specifications of the AR Drone 2 are (Irizarry et al., 2012):

- HD (high definition) Video recording and live video streaming to the smartphone or tablet as a pilot is flying along with clean, sharp image just as if you are in the pilot seat with the resolution of 720p 30fps. Mounted with wide angle 92° diagonal liquid lens along with H264 encoding base profile and low latency streaming. The video storage on the fly is with the remote device as iPad via Wi-Fi and the image format is JPEG.
- Robust structure with Carbon fiber tubes and total weight of 380g with outdoor hull, 420g with indoor hull. High grade 30% fiber charged nylon plastic parts foam to isolate the inertial center from the engines' vibration and EPP hull injected by a sintered metal mold. It is also equipped with liquid Repellent Nano-Coating on ultrasound sensors.
- Fully reparable: All parts and instructions for repairing available on the internet Electronic assistance
- AR.Drone 2.0 on-board technology gives you extreme precision control and automatic stabilization features.
- It has 1GHz 32 bit ARM Cortex A8 processor with 800MHz video DSP TMS320DMC64x and Linux 2.6.32 OS with 1Gbit DDR2 RAM at 200MHz and USB 2.0 high speed for extensions along with Wi-Fi b,g,n classes and 3 axis gyroscope 2000°/second precision.
- Mounted with 3 axis accelerometer +/-50mg precision, 3 axis magnetometer 6° precision, Pressure sensor +/- 10 Pa precision, Ultrasound sensors for ground altitude measurement, 60 fps vertical QVGA camera for ground speed measurement for the motor to fly high, fast or far away from the ground.
- It has 4 brushless in-runner motors (14.5W 28,500 RPM) with Micro ball bearing Low noise Nylatron gears for 1/8.75 propeller redactor.
- Tempered steel propeller shaft Self-lubricating bronze bearing, specific high propelled drag for great maneuverability, 8 MIPS AVR CPU per motor controller, 3 elements 1000 mA/H LiPo rechargeable battery (Autonomy: 12 minutes), emergency stop controlled by

software, fully reprogrammable motor controller, water resistant motor's electronic controller.

2. METHODOLOGY

By flying with the Ar. Drone 2 and capturing 14 movies from the area in the test bed and finally using Free Video to JPEG convertor, many images were created for the project. Since the AR drone 2.0 has liquid wide angle lens, in the movies there are lots of deformations and distortions and it is difficult to do image matching based on Epipolar lines for automatic 3D modelling. The movies were converted to jpeg files and then we have tested different applications such as AgiSoft Photoscan or Autodesk Recap 360 for automatic 3D modelling but the result was not acceptable due to lack of accuracy and quality (See Figures 3,4). In AgiSoft PhotoScan it is possible to do camera calibration based of images and follow the workflow step by step such as adding photos, aligning photos, building dense cloud, building mesh, building texture and etc. The final 3D format also can be edited in AgiSoft PhotoScan software by means of select or crop tools or can be exported in 3DS, WRL, COLLADA, KMZ, DXF and many other formats. In this project we have exported the 3D model in 3DS format and imported to SketchUp Trimble for the editing purpose. Another problem is the obstacles such as trees and parked vehicles on the road or in front of the building which are modelled in a very low quality connected to the 3D building. It is boring to solve these problems manually and this the main reason that a Ruby programming languages were used to develop a plugin for the SketchUp Trimble for automatic modelling purpose. In the first step movies are converted to jpeg files based on 10 fame per second and then the pictures analysed and parallel projection a redeveloped method which is called Dynamic Pulse Functions (DPF) (Alizadehashrafi, 2015) has been enhanced and used to create 3D models in LOD3 via Ruby programming language (Scarpino, 2010).

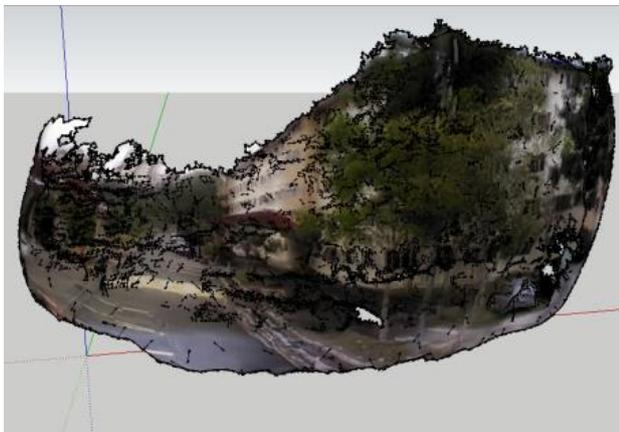


Figure 3. Low quality 3D model before editing, extracted from AgiSoft PhotoScan in 3DS format and imported to SketchUp Trimble.



Figure 4. Low quality 3D model along with all the obstacles such as trees and vehicles after editing via SketchUp Trimble.

3. RUBY CONSOLE IN SKETCHUP TRIMBLE

The concept of DPF can be enhanced and utilized for the windows layer along with extra parameter of deepness of the windows for each layer and also the raised shade of the windows in upper part of them in UA1 and UA2 buildings in Kolej 9, UTM, Skudai. In this case, all the windows geometries on the façade can be generated automatically by coding the plugin and adding to the menu bar. The following Ruby code can represent the concept of DPF by detail. In the first step the parameters were measured from the rectified photo after parallel projection. The photos also are extracted from the movies of Ar Drone 2.0 via Free Video to JPEG convertor software. These parameters are height and width of the building, number of similar windows in each floor and layer, deepness of the windows, number of floors or levels, the height and width of the window in each layer, horizontal and vertical distances between two windows in each layer, the upper shade of the windows, vertical and horizontal starting point from upper left corner for the windows layer, the texture file name and path (See Figures 6 and 7). The complexity of the program is $O(n^2)$ as its using two nested loops to create one layer of window automatically. The file in '.rb' format along with the 'win' folder which is highlighted in the following code and contains the image texture file for the windows layer, were zipped as a 'zip' file. Then the extension renamed from 'zip' to 'rbz' to be installed as a plugin in SketchUp Trimble for automatic 3D modelling process (See the appendix).

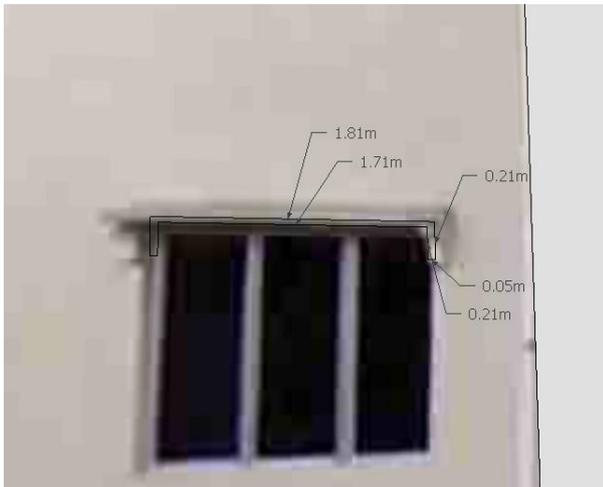


Figure 5. Parameters of DPF for automatic 3D modelling

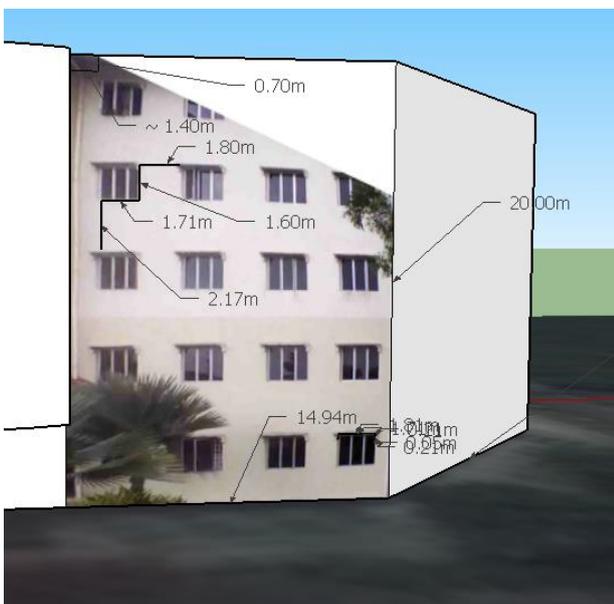


Figure 6. Parameters of DPF for automatic 3D modelling

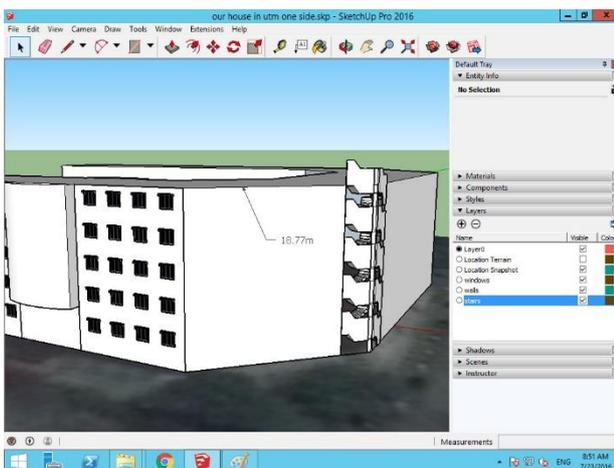


Figure 7. Width of the façade for automatic 3D modelling in LoD3.

4. RESULTS

The 3D models generated in LoD3 with the concept of DPF from the rectified photos extracted from the movies of Ar drone 2.0. The quality of the 3D models is very high with detailed geometries of windows and walls and the data size is roughly 6.5MB. The model uploaded to Trimble 3D warehouse and it is available for downloading (Alizadehashrafi).

ACKNOWLEDGEMENTS

We would like to appreciate the research grant from both universities of the authors for this project.

REFERENCES

- AGISOFT, L. 2014. Agisoft photoscan. Professional Edition.
- ALIZADEHASHRAFI, B. UA1 and UA2 in Kolej 9 UTM, Skudai.
<https://3dwarehouse.sketchup.com/model.html?id=62528018-88b8-49f3-982c-95da8c6adc29>.
- ALIZADEHASHRAFI, B. 2015. Using Parameters of Dynamic Pulse Function for 3d Modeling in LOD3 Based on Random Textures. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 51-55.
- IRIZARRY, J., GHEISARI, M. & WALKER, B. N. 2012. Usability assessment of drone technology as safety inspection tools. Journal of Information Technology in Construction, 17, 194-212.
- SCARPINO, M. 2010. Automatic SketchUp: Creating 3-D models in ruby, Eclipse Engineering LLC.
- STRAUB, J. & KERLIN, S. Evaluation of the use of 3D printing and imaging to create working replica keys. SPIE Defense+ Security, 2016. International Society for Optics and Photonics, 98230G-98230G-14.

APPENDIX

```
require 'sketchup.rb'
UI.menu("Plugins").add_item("Create Windows") {
  createWindows
}
def createWindows
  hightFacade = 20.m
  widthFacade = 18.77.m
  numberOfWindows = 5
  deep = -0.15.m
  raised = 0.25.m
  levels = 5
  windowWidth = 1.71.m
  windowHeight = 1.60.m
  horizontalCluster = 1.80.m
  verticalCluster = 2.17.m
  horizontalDistanceUpperLeftCorner = 2.m
  verticalDistanceUpperLeftCorner = 0.70.m
  winShadeHeight = 0.21.m
  winShadeWidth = 0.05.m
end
```

```
model = Sketchup.active_model
entities = model.entities
wall= entities.add_face [widthFacade,0, 0], [widthFacade,
0,heightFacade ], [0,0,heightFacade],[0,0,0]
z = heightFacade - verticalDistanceUpperLeftCorner -
windowHeight
for step in 1..levels
x = horizontalDistanceUpperLeftCorner
for step in 1..numberOfWindows
x += windowWidth
y = 0
pt1 = [x, y, z]
pt2 = [x, y, z + windowHeight]
pt3 = [x - windowWidth, y, z + windowHeight]
pt4 = [x - windowWidth, y, z ]
ptShade1 = [x+0.01.m, y, z + windowHeight+0.01.m]
ptShade2 = [x+0.01.m, y, z + windowHeight-winShadeHeight]
ptShade3 = [x+winShadeWidth, y, z + windowHeight-
winShadeHeight]
ptShade4 = [x+winShadeWidth, y, z +
windowHeight+winShadeWidth]
ptShade5 = [x-winShadeWidth-windowWidth, y, z +
windowHeight+winShadeWidth]
ptShade6 = [x-winShadeWidth-windowWidth, y, z +
windowHeight-winShadeHeight]
ptShade7 = [x-windowWidth-0.01.m, y, z + windowHeight-
winShadeHeight]
ptShade8 = [x-windowWidth-0.01.m, y, z +
windowHeight+0.01.m]
x += horizontalCluster
WINDOW = entities.add_face pt1, pt2, pt3, pt4
WINSHADE = entities.add_face ptShade1, ptShade2, ptShade3,
ptShade4,ptShade5, ptShade6, ptShade7, ptShade8
imdadr = Sketchup.find_support_file "/win/win.jpg", "Plugins"
mats = Sketchup.active_model.materials
nwmat = mats.add "nmat1"
nwmat.texture = imdadr
WINDOW.material = nwmat
pts = []
pts[0] = Geom::Point3d.new(pt1)
pts[1] = Geom::Point3d.new(1,0,0)
pts[2] = Geom::Point3d.new(pt2)
pts[3] = Geom::Point3d.new(1,1,0)
pts[4] = Geom::Point3d.new(pt3)
pts[5] = Geom::Point3d.new(0,1,0)
pts[6] = Geom::Point3d.new(pt4)
pts[7] = Geom::Point3d.new(0,0,0)
WINDOW = WINDOW.position_material(nwmat,pts, true)
WINDOW.pushpull deep
WINSHADE.pushpull raised
end
z -= windowHeight
z -= verticalCluster
end
groupWINDOWS = ent.add_group WINDOW.all_connected
```