ISSN 2278-3091

Volume 9, No.1, January – February 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse23912020.pdf

https://doi.org/10.30534/ijatcse/2020/23912020

# Simulating the Effect of Non-Axial Airflow on the Motion of an Undriven Propeller

**Francis Gregory Ng[1], Alvin Chua[2]**

Mechanical Engineering Department, De La Salle University - Manila, Philippines, francis_gregory_ng@dlsu.edu.pl

[2]Mechanical Engineering Department, De La Salle University - Manila, Philippines, alvin.chua@dlsu.edu.ph

## ABSTRACT

It is evident that a windmill propeller action can be induced by airflow along its rotation axis. Whereas it has been proven that a spinning propeller has a significantly lower drag than a fixed one, this is unlikely to be the case when the airflow is non-axial. To properly account for this type of airflow, a series of flow simulations has been done. Photogrammetry has also been employed since it is essential to accurately capture the geometry of an actual propeller and avoid any oversimplification of the model. As a result of this investigation, it was found that a stable orientation is achieved when the length of the propeller is roughly perpendicular to the airflow. An undriven propeller is naturally static when exposed to non-axial airflow and thus when this scenario is part of a system, further flow analysis need not be transient.

**Key words :** computational fluid dynamics, photogrammetry, propeller, simulation.

## 1. INTRODUCTION

The separate-lift-and-thrust (SLT) hybrid drone [1] is a practical development which has increased utility compared to traditional fixed-wing and quadcopter drones. As a combination of these two formats, it is capable of vertical take-off and landing, as well as sustained straight and level flight. A major concern, however, is the penalty on the aircraft performance caused by the added components. To quantify this, the equation in [2] is used and this requires the calculation of the drag.

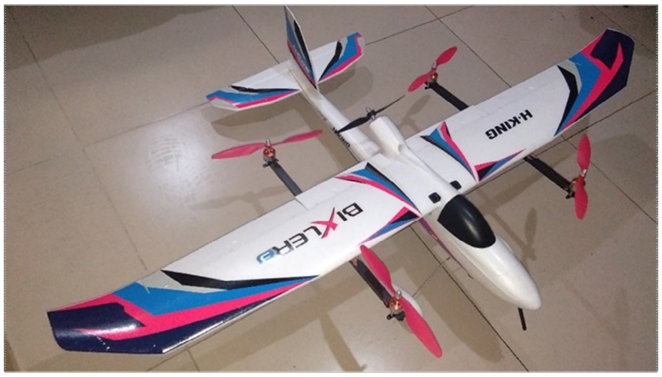

**Figure 1:** Separate-Lift-and-Thrust Hybrid Drone

Flow simulations using computational fluid dynamics (CFD) software are quite useful in obtaining the drag calculations. But in utilizing this rather than a wind tunnel, accurately incorporating the propeller into the simulation scenario will prove to be an important task. Even though the consideration for drag is straight and level flight (i.e. the vertical thrust propellers are not driven), there might be a possibility that the propellers will spin because of the relative airflow.

The effect of propeller spin can be significant since a study shows that the drag increases substantially when a propeller is fixed rather than allowed to spin freely [3]. Yet, this is inapplicable to the hybrid drone scenario where the fluid flow is not axial; the basic intuition that stationary propellers will spin according to the flow does not necessarily apply. Hence, it is important to determine how the propeller will actually react to non-axial airflow. With this study, the prospect is explored by means of flow simulations coupled with a methodology to create fairly accurate propeller geometry.

## 2. CAD MODELING

### 2.1 Photogrammetry

Computer-aided design (CAD) is a prerequisite to working with CFD. However, it would be difficult to make a CAD model of the propeller parametrically through measurements of the actual object. The propeller twist is not constant and has been designed for specific mission conditions [4]. So to successfully create a CAD model, 3D scanners may be utilized. The issue with this choice, however, is the cost which may make the methodology challenging to replicate.

Photogrammetry is an effective alternative which reconstructs a 3D mesh based on images from multiple angles. Reconciling multiple images is not a unique proposition since [5] does this in a medical imaging context but the distinction with photogrammetry is the goal of forming a 3D surface. This starts with structure from motion (SfM) or also called the multi-view geometry (MVG) which includes features extraction, camera orientation recovery, and sparse point cloud generation. This is followed by multi-view stereo (MVS) which generates the dense point cloud and the mesh reconstruction. Manual mesh clean-up may subsequently be required depending on the quality of pictures and also the software implementation.
Photogrammetry is a mature field with already more than 80 different software available [6]. Not all of these must be exclusively purchased, thus users such as [7] have been able to test these free photogrammetry software options. In addition, a concrete reference is [8] who conducted a more elaborate test with reference to ground-truth coming from high-quality 3D laser scans.

Both the above references assisted in deciding between the possible software choices in terms of compatibility and effectiveness. There are actually a multitude of commercial photogrammetry software available, but these mostly require hefty subscription prices even for educational purposes or otherwise impose significant restrictions. Thus, these are not considered here.

For the MVG stage of photogrammetry, *VisualSfM* [9] can be used but it is relatively old and has little modifiable parameters. Alternatives are *OpenMVG* [6] and *OpenSfM* [10] which are all relatively more up-to-date and have more comprehensive settings.

For the MVS stage, the most extensive option appears to be *OpenMVS* [11]. In contrast, *MeshRecon* [12] seems rather basic in its usage, and *CMVS* [13] is around as old as *VisualSfM*. Some alternatives also span the whole photogrammetry pipeline (both *MVG* and *MVS*). These include *COLMAP* [14] and *MVE* [15]. Another option is *ReCap* [16] but it is only free for students. Though black-box due to its commercial nature, the advantage of *ReCap* is its cloud processing capability. Thus, it does not require much local resources but has limitations in cloud storage and the number of pictures it can process.

For this study, *OpenMVG* and *OpenMVS* were chosen. Since both are open-source, these are readily available for the community to assess much like the manner of [17]. This also means that executables can be built from the source. Version-compatibility issues with dependencies would prove to be challenging and this may be discouraging to accomplish. Nevertheless, build instructions are available, and if needed, the old versions of the executables themselves can be directly retrieved instead of the source.

When the executables are obtained, a single Python script can be used to handle batches of photogrammetry datasets in sequence. This avoids the need for repetitive user interaction. With the script, images that are simply placed into labelled folders and all the output are compiled into a single folder with proper identification.

Next, to obtain the images for the reconstruction, a photogrammetry setup is needed. In this study, a budget camera with manual settings was used and set at an aperture of f/8, a shutter speed of 1/100, and an ISO 100 setting. Since the environment lighting could affect the exposure, the shutter speed is slightly adjusted to compensate.

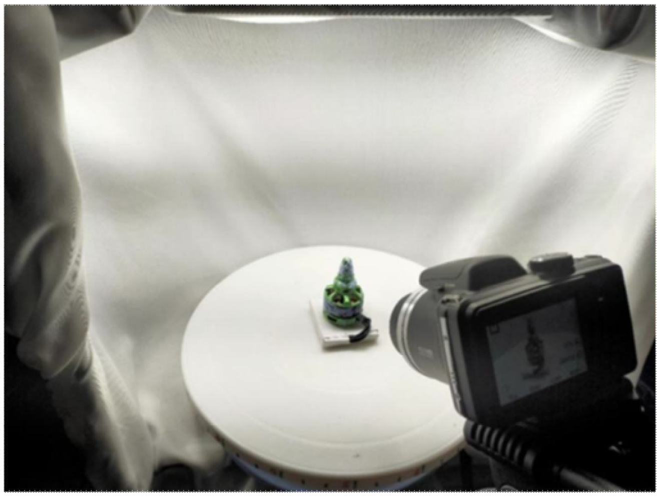

**Figure 2:** Photogrammetry Setup

The high f-number increases the depth of field to keep most of the subject in focus, the fast shutter speed avoids motion blur (a camera on a tripod may still shake when the button is pressed), and the low ISO setting reduces noise [18], [19]. A cellphone camera may be used if the image quality is good in these aspects. Caution must be taken, though, to avoid post-processing such as in [20] and [21] since the pixel changes may possibly affect the reconstruction process.

With these camera settings, the photos become less sensitive to lighting, thus requiring a bright setup to achieve good exposure. For this, a small photography lightbox is constructed by mounting small floodlights onto a structure made of extra shelf materials. The light is diffused using plain cloth, helping to reduce hard shadows which the software may mistakenly identify as object features. The cloth also serves as a featureless backdrop.

Lastly, to complete the lightbox, a small white turntable is used which works well with the camera tripod to systematically obtain revolutions of photos. The turntable has markings for reference in maintaining a roughly constant rotational displacement between photos. Then, adjusting the tripod height allows for capturing photos from a different orbit. In contrast with professional photogrammetry stations such as that used by [19], the physical setup in Figure 2 is limited to small-object photogrammetry.

### 2.2 Procedure

For this study, the motor was reconstructed alongside the propeller to illustrate the procedure. Nevertheless, the main focus is the creation of the propeller CAD model for use in the CFD analysis. As previously mentioned, a propeller has a twisting geometry. This varies the pitch angle throughout the length of the blade and renders it difficult to capture through manual measurement or orthographic projections. Also, the fact that the sections of the blade are asymmetrical further adds complexity. This warrants the use of the photogrammetry setup discussed in the previous subsection.

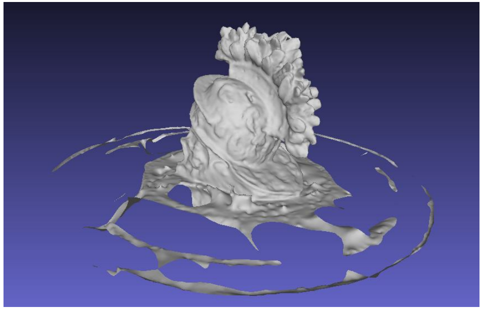

**Figure 3:** Trial photogrammetry using miniature helmet

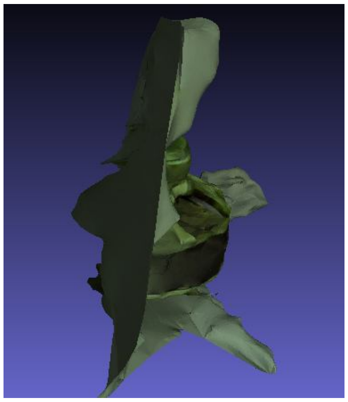

**Figure 4:** Failed photogrametry reconstruction of a motor

Before the photogrammetry lightbox was finalized, initial trial runs were done on a miniature helmet (Figure 3). The results appeared quite acceptable. The crest of the helmet has actually been captured quite extraordinarily. Even though the turntable is partially polluting the mesh, this can be cleared along with the other imperfections. As the object is much more complicated than a quadcopter motor and has greater luster that complicates the photogrammetry, it was expected to be a good test subject. However, the reduced complexity of the motor is actually a drawback. With flatter features that are circularly repetitive, the software has difficulty in recreating the object as shown in Figure 4.

Figure 5 shows a successful reconstruction of the motor where the features have been accentuated by the application of random markings. This approach was inspired by the process of spray coating reflective surfaces for photogrammetry which made it possible to use an unsuitable transparent surface [19]. However, the reliability of simply adding markings is not good enough. In some cases, the markings are even inapplicable to the motor and propeller surfaces.

Thus, a superior alternative has been found in applying tack (adhesive putty) onto the surface of the motors and propellers. This better aided in increasing features for successful software processing as there is assured visibility and application, as well as some added texture. Without the random textures from the tack, the plain and repetitive design of these components would make it difficult to reconstruct the meshes of the components. Practically, the texture is not outright detrimental since this can later be smoothened out.

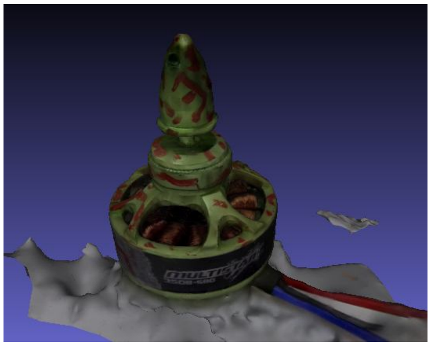

**Figure 5:** Successful photogrammetry reconstruction of a motor due to additional markings

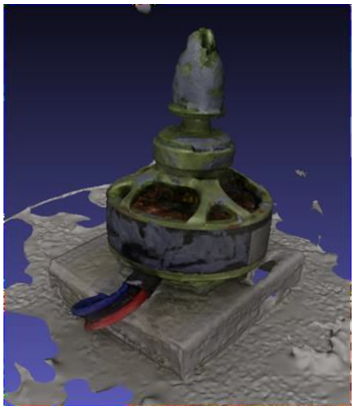

**Figure 6:** Successful photogrammetry reconstruction of a motor due to applied tack

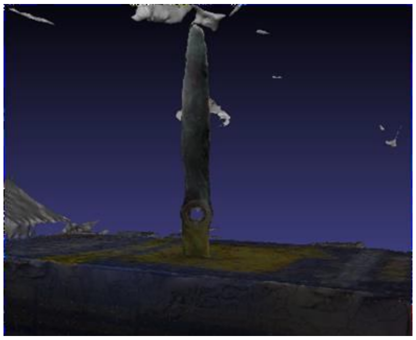

**Figure 7:** Successful photogrammetry reconstruction of a propeller blade due to applied tack

Figures 6 and 7 show the reconstructed motor and propeller, respectively. A small box hides extra wiring from the motor. Similarly, a large box helps support the propeller upright; otherwise, if the propeller lays flat on the turntable, one side of each blade will be out of view. An additional benefit of simply orbiting around a single blade is that more common points are maintained between images. Also, symmetry in the final output is ensured as the remaining part of the propeller will be a revolved copy of the reconstructed blade.

For the choice of CAD software, instead of using *SolidWorks*, a good alternative is *Autodesk Fusion 360* because it is free for educational and small-scale commercial use. Having the mesh results from the photogrammetry, these are resized and unwanted faces are deleted. Then, T-splines are used to wrap around the mesh to smoothen the geometry. The meshes themselves may possibly be used but irregularities may be a problem. Thus, to ensure better representation, these are instead used as guides in constructing the final CAD models of the components.

As shown in Figure 8 below, the mesh body is used as the basis for a T-spline. This approach smoothens out the roughness from the tack, also practically interpolating any missing faces from faulty reconstruction and cleaning. The process is done by using the pull function for bringing the control points of a cylindrical T-spline toward the mesh surface. After some adjustments and the closure of the T-spline, a solid blade body is formed. The propeller is completed by mirroring the blade and adding a hub. From here, a propeller of the opposite spin direction may be mirrored. This is best done using an upright reference plane passing along a propeller blade to avoiding any need for reorientation.

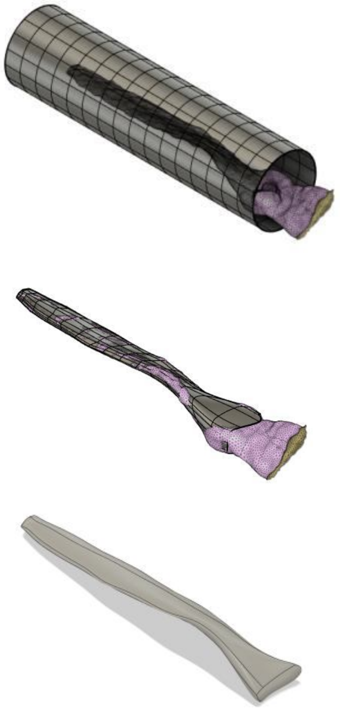

**Figure 8:** T-spline modeling of a propeller blade from coarse photogrammetry mesh

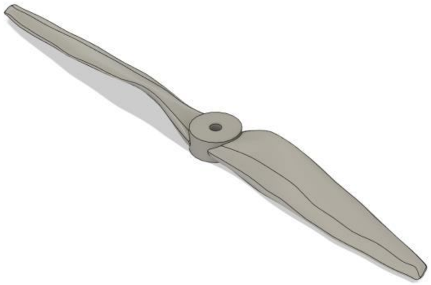

**Figure 9:** Completed CAD model of a propeller

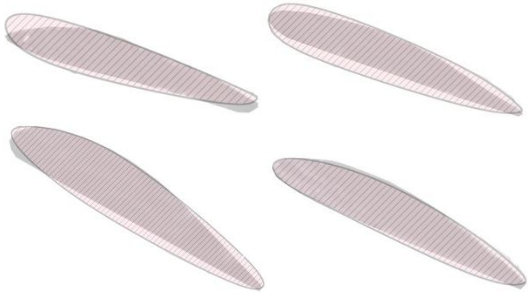

**Figure 10:** Section analysis of a propeller.

To verify the quality of the reconstruction, a simple section analysis is done on the propeller CAD which is used later in the simulation. For comparison, section images were obtained by cutting the actual propeller and these were overlaid as the dark shades in the figure. Clockwise from the top left, are the sections at 25 mm, 50 mm, 75 mm, and 100 mm from the tip of the blade. The respective errors in cross-sectional area are 16.46%, 41.53%, 4.71%, and 18.37%.

Discrepancies in matching the leading curvature can be observed, as well as trouble in reconstructing a sharp trailing edge. This can be improved by adding more T-spline vertices but may result in less smoothening effect intended for counteracting the application of tack in the photogrammetry. Therefore, to improve the reconstruction quality, it may be necessary to use a less intrusive alternative to tack for introducing irregularities in the object surface. Nevertheless, since the reconstruction adequately matches the size and twist of the actual propeller, as well as the general shape of the cross-sections, it is already sufficient for use in this study.

## 3. CFD SIMULATION

For the flow simulation, the largest propeller on hand was chosen and subjected to 50 kph of wind. The propeller is an 11×7 and it has been split into two halves to enable the wall calculator to independently evaluate either blade (Figure 11). This is done in *Autodesk CFD Ultimate*. Like *Fusion 360*, this is free for academics for a substantial period of time, making it a practical choice.

To generate the computational mesh in the CFD, automatic sizing is done with some user-provided parameters which are the wall layer settings and the size adjustment. The wall layer is set to 10 layers, the layer factor to 0.6, and the layer gradation to 1.25, as advised for the shear stress transport (SST) k-omega turbulence model, which is more appropriate for external aerodynamics than the default general-purpose k-epsilon model. This model is more accurate in the boundary layer and considers the effect of SST which is important for separation predictions [22]. While a hybrid approach with Large Eddy Simulation (LES) can predict more complex flow structures, it is not necessarily superior in predicting the absolute drag value and requires more computational cost [23]. Hence, the choice of using the SST k-omega turbulence model is quite appropriate for this study.

The mesh size can be adjusted based on a factor from 0.2 to 5. This parameter indicates the factor of size increase from the original generated mesh. The size adjustment factor used is 0.3 based on results from a mesh independence test. It can be noted that the software has embedded mesh quality constraints which override the adjustments whenever the mesh is excessively coarsened. Nevertheless, this is not quite relevant as a substantial degree of fineness is necessary to obtain accurate results.

## 4. RESULTS

With air flowing directly from the top (into the propeller shown in Figure 11), the calculated forces on the blades are shown in the Table 1. This would result in a clockwise motion, as expected, if there is no starting torque requirement. The software calculates the moment to be -0.0189403 N-m. The sign of this value matches the right-hand rule with the Cartesian plane overlaid on the above image. This coordinate system is exactly what was used in the simulations and thus, for subsequent references to the propeller orientation, the angle is denoted by the position of the right blade (red), which currently sits at 0°.

For the SLT hybrid, the relative wind is non-axial, largely moving along the positive y-direction. Predictably, the resulting forces in both blades are oriented with the wind, thus having a cancelling effect. Still, a small differential exists, presumably due to the fact that wind is diverted more effectively by the side of the propeller blade which was designed to generate the downwash, as could be seen in the mismatch of generated lift ($F_z$) Nevertheless, this rotational force is a magnitude smaller from the axial wind scenario having resulted in significantly smaller calculated moment of 0.00106691N-m. Noting the sign of this value, the propeller actually spins in the counter-clockwise direction because the higher lift, hence the greater induced drag, is on the right blade (red).

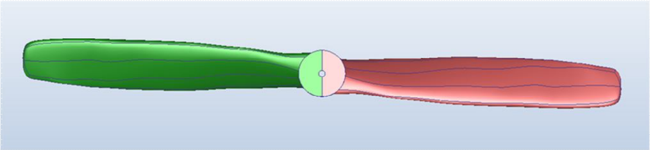

**Figure 11:** Propeller used in CFD simulation.

**Table 1:** Forces caused by axial airflow into the propeller

| Blade | $F_x$ (N) | $F_y$ (N) | $F_z$ (N) |
|---|---|---|---|
| Left (green) | 0.00467292 | 0.132045 | -0.365872 |
| Right (red) | 0.00507012 | -0.133714 | -0.369101 |

**Table 2:** Forces caused by non-axial airflow to the propeller (perpendicular to the length of the blade)

| Blade | $F_x$ (N) | $F_y$ (N) | $F_z$ (N) |
|---|---|---|---|
| Left (green) | -0.00345916 | 0.144753 | -0.165514 |
| Right (red) | 0.0136067 | 0.157279 | 0.212793 |

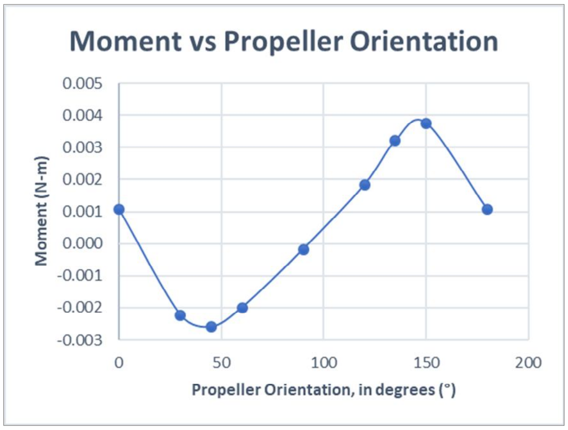

**Figure 12:** Variation of Moment on the Propeller at Different Orientations Relative to the Non-Axial Airflow.

To ensure that a spin will be produced, a moment needs to exist given an unbalanced force. Additionally, this moment has to be sustained. As non-axial flow is not the design for the propeller, the forces may vary while the propeller moves, even if the wind is constant. In fact, the above scenario considers the maximum lift that a blade can produce but does not exactly cover the case where the moment is largest. For example, when the propeller is rotated counter-clockwise by 45°, this value somehow increases to -0.00259913 N-m, a different spin direction. But when it turns more to 90°, this drastically diminishes to only -0.000174259 N-m. Therefore, it becomes imperative to simulate the propeller at different orientations and analyze its behavior, leading to Figure 12 which can be interpreted rather clearly.

The moment on the propeller is periodic for every 180° rotation. This was not further divided into 90° since there is an effect for which side of the propeller blade is advanced into the wind. In the figure, some minor asymmetry arises such as the difference in the peak magnitudes for the moment, so the plot is distorted from being a pure sinusoid. Nevertheless, with this behavior, it is possible to deduce the stable position of the propeller. The moment becomes zero at roughly the 0° and 90° orientations.

Recalling the right-hand convention, a positive value would result in an increase of the angle and the reverse if the value were negative. The implication is that a stable solution must have a negative slope, and this is satisfied only by the solution at approximately 0°, which is the initial orientation shown in Figure 11 and discussed at the start of this section.

## 5. CONCLUSION

With this test, it can be concluded that the propeller can practically be driven only by axial flow. This brings to mind the contrast between horizontal axis wind turbines and vertical axis wind turbines. Propellers only belong to the former since the axis is aligned to horizontal wind. Reorienting such a turbine would not immediately make a functional vertical axis turbine.

Appropriately, the design of the quadcopter propellers is only limited for axial airflow. No conscious effort could have been made to allow it otherwise as there is no associated practical purpose. Hence, driven motion would take place only when the drone's angle of attack is large enough to give a substantial axial component of the relative speed. The axial wind component at reasonable pitch angles would be expectedly quite low and may not move the propellers because of the resistive torque present in the motors.

As a conclusion from this investigation, propeller rotation would not be considered in the flow simulations of the SLT hybrid drone as it is expected to remain stationary. Transient analysis is unnecessary for a steady-state scenario, allowing for reduced simulation complexity. The proper orientation of the propeller in such an analysis has also been set by this study. Therefore, calculating the efficiency of the hybrid will be much more obtainable.

## ACKNOWLEDGEMENT

The authors would like to acknowledge the scholarship support provided by the Department of Science and Technology – Science Education Institute (DOST-SEI) and the Engineering Research and Development for Technology (ERDT).

## REFERENCES

1. P. Dewi, G. Hadi, M. Kusnaedi. A. Budiyarto, and A. Budiyono. **Design of Separate Lift and Thrust Hybrid Unmanned Aerial Vehicle**, in *The Journal of Instrumentation, Automation and Systems*, Vol. 2(2), 2015, pp. 45-51. https://doi.org/10.21535/jias.v2i2.697
2. L. Traub. **Range and endurance estimates for battery-powered aircraft**, in *Journal of Aircraft*, Vol. 48(2), 2011, pp. 703-707.
3. P. MacKenzie and M. Forrester. **Sailboat propeller drag**, in *Ocean Engineering*, Vol. 35(1), 2008, pp. 28-40.
4. S. Gudmundsson. ***General Aviation Aircraft Design***: Butterworth-Heinemann, 2014, ch. 14, pp. 581-659. https://doi.org/10.1016/B978-0-12-397308-5.00014-3
5. C. Uma, M. Srinivasa, S. Ram, K. Babulu, and D. Prakash. **Image Registration of Multi Model Enrollment using Mutual Information Technique**, in *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 7(6), 2018, pp. 107-110. https://doi.org/10.30534/ijatcse/2018/07762018
6. P. Moulon, P. Monasse, R. Perrot, and R. Marlet. **OpenMVG: open multiple view geometry**, in *International Workshop on Reproducible Research in Pattern Recognition*, 2016, pp. 60-74.
7. P. Falkingham. **Free photogrammetry software review: 2017**, [Online]. Available: https://peterfalkingham.com/2017/12/17/free-photogrammetry-software-review-2017/ [Accessed: 17-Oct-2018].
8. A. Knapitsch, J. Park, Q. Zhou, and V. Koltun. **Tanks and temples: benchmarking large-scale scene reconstruction**, in *ACM Transactions on Graphics*, Vol. 36(4), 2017, pp. 78:1-78:13. https://doi.org/10.1145/3072959.3073599
9. C. Wu. **VisualSFM: A visual structure from motion system**, [Online]. Available: http://ccwu.me/vsfm/ [Accessed: 17-Oct-2018].
10. Mapillary. **OpenSfM**, [Online]. Available: https://docs.opensfm.org/index.html [Accessed: 26-Oct-2018].

11. OpenMVS authors. **OpenMVS**, [Online]. Available: https://github.com/cdcseacave/openMVS/wiki [Accessed: 26-Oct-2018].
12. Z. Kang. **Mesh reconstruction from imagery**, [Online]. Available: http://zhuoliang.me/meshrecon.html [Accessed: 26-Oct-2018].
13. Y. Furukawa. **Clustering views for multi-view stereo (CMVS)**, [Online]. Available: https://www.di.ens.fr/cmvs/ [Accessed: 26-Oct-2018].
14. J. Schoenberger. **COLMAP**, [Online]. Available: https://colmap.github.io/index.html [Accessed: 17-Oct-2018].
15. GCC - Technische Universität Darmstadt. **Multi-view environment**, [Online]. Available: https://www.gcc.tu-darmstadt.de/home/proj/mve/ [Accessed: 17-Oct-2018].
16. Autodesk Inc. **ReCap**, [Online]. Available: https://www.autodesk.com/products/recap/overview [Accessed: 26-Oct-2018].
17. C. Dim, F. Nabor, G. Santos, M. Schoeler, and A. Chua. **Novel Experiment Design for Unmanned Aerial Vehicle Controller Performance Testing**, in *IOP Conf. Series: Materials Science and Engineering*, Vol. 533(1), 2019. https://doi.org/10.1088/1757-899X/533/1/012026
18. 3Dflow. **3DF Zephyr photography guide**, [Online]. Available: https://www.3dflow.net/technology/documents/photogrammetry-how-to-acquire-pictures/ [Accessed: 30-Sep-2019].
19. J. Busby. **3D scanning reflective objects with photogrammetry**, [Online]. Available: https://www.3dscanstore.com/blog/3d-scanning-reflective-objects [Accessed: 30-Sep-2019].
20. N. Leelavathy, V. Kumari, K. Jyotsna, P. Nagamani, and B. Kishore. **A Relative Study of Several Techniques for Underwater Image Improvement and Restoration**, in *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 7(6), 2018, pp. 115-119.
21. P. Babu. **Super Resolution Image Reconstruction for single image using Approximate BPTSRIRTD Algorithm**, in *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 7(6), 2018, pp. 144-148. https://doi.org/10.30534/ijatcse/2018/16762018
22. F. Menter. **Review of the shear-stress transport turbulence model experience from an industrial perspective**, in *International Journal of Computational Fluid Dynamics*, Vol. 23(4), 2009, pp. 305-316.
23. C. Zhang, C. Bounds, L. Foster, and M. Uddin. **Turbulence modeling effects on the CFD predictions of flow over a detailed full-scale sedan vehicle**, in *Fluids*, Vol. 4(3), 2019, pp. 1-28. https://doi.org/10.3390/fluids4030148