

Towards a Holistic Performance Evaluation Framework for Drone-Based Object Detection

P. Petrides, C. Kyrkou, P. Kolios, T. Theocharides and C. Panayiotou
KIOS Research Center for Intelligent Systems and Networks
Department of Electrical Engineering, University of Cyprus
Nicosia, Cyprus

Abstract—Recent advances in drone visual sensors and integration of complex vision algorithms, facilitate further potential, entirely disrupting in a positive way their applications and capabilities. In particular, real-time object detection, usually the initial necessary step in multiple computer vision and image processing applications, has been gaining momentum in drone-based applications. Whilst heavily researched in conventional systems, drone-based vision algorithms have to consider extrinsic parameters to measure their efficiency, as their performance is heavily impacted by various flying parameters such as altitude. Further, the parameters that directly impact the performance of the vision algorithms, also impact the duration of the flight (i.e. battery life), as the vision algorithmic performance is affected by the flying route and altitude as well. This paper therefore, presents a holistic performance evaluation framework for multi-rotor drone-based object detection applications, that considers various trade-offs such as flight duration, camera resolution, computational platform performance, drone battery performance, etc., in providing a thorough analysis of the various factors affecting the operation of object detection. The framework showcases indeed that the flying altitude, in combination with the camera resolution, vastly impacts the flight duration as well as the performance of the object detection algorithm, when targeting coverage of a specific area. The framework has been experimentally verified using a commercial grade state-of-the-art drone and high-resolution camera, as well as a high-end embedded processing platform that performs the detection algorithm.

I. INTRODUCTION

Commercial multi-rotor UAS have become ubiquitous in our daily lives, and are expected to dominate the consumer electronics market in an unseen trend, as their application spectrum ranges from civil protection, to entrepreneurial activities, to hobbyists and many more. In the recent past, the use of UAS has been mainly focused in military and high end aerospace applications requiring enormous economical investments. Nowadays, such a technology is widely available to anyone at very low cost. By doing a spherical overview of the capabilities of these autonomous flying machines, someone can easily conclude that UAS can be beneficiary to a wide range of applications including search and rescue, intelligence surveillance, traffic monitoring and inspection, disaster evaluation, wildfires monitoring, border patrolling, photography and media productions etc. They are now equipped with high-end sensing and communication nodes, and low-power lightweight on-board embedded processing

platforms, while they also exhibit increased flying times, excellent maneuverability, and extended ranges of operation. Their ability to carry a variety of sensors and payload while being able to monitor from above, give a huge advantage to the ground units in terms of cost, situational awareness as well as safety by minimizing the use of manned helicopters.

Introducing computer vision and specifically object detection in UAS can be beneficial in a broad range of applications. For the purposes of this paper rotorcrafts will be considered at all times. Automatic object detection either real time or offline is a very hot subject for computer vision researchers. Already, an extensive research has been done in the part of object detection algorithms and the outcome seems to be very promising for a wide range of applications involved in automated systems. Real time aerial object detection adopted in UAS give an intelligent autonomous flying machine that is suitable for an enormous number of applications. Since commercial low priced UAS technology is widely available, aerial object detection is getting more and more popular. At very low cost, it is possible to have a complete autonomous vision based aerial system.

The purpose of this paper is to investigate how different parameters of flight performance can affect object detection algorithms and vice-versa. It will be shown that parameters such as flight altitude, overall weight, camera specifications are all interrelated with each other and affect directly or indirectly the performance of an object detection algorithm as well as the total flight time. The flight time is an important parameter for any application, especially when path planning algorithms are being used. With current technology available, batteries have limitations in terms of capacity versus weight meaning that is the main drawback. Many researchers across the world are investigating how to minimize power consumption of embedded systems in UAS in order to counterbalance the battery capacity limitation. However, this paper will demonstrate that power consumption from inbound electronics is not only the case but weight is much more important factor for maximizing overall flight time

II. BACKGROUND

Computer vision has been mainly introduced into military UAS in the past with extremely high costs, however this

technology has been introduced recently in commercial UAS. Since then, the combination of computational vision and unmanned flying machines has been investigated in a variety of aspects. For instance, power line inspection is a difficult and risky task for the maintenance personnel, however, an autonomous vision-based power line inspection using UAS could be extremely beneficiary as introduced in [1], [2]. An application that a fully autonomous UAS will be able to provide a massive help, is Search and Rescue. During such operation parameters such as time management, response time and situational awareness are crucial. A fully autonomous vision-based UAS can monitor a search area and provide useful aerial information to first responders. In [3], a UAS carrying thermal and optical cameras is executing a scan into Search and Rescue field while at the same time is performing image processing to detect human bodies. A multi-purpose use of the combination of vision and UAS is simultaneous localization and mapping as well as obstacle avoidance system. The latter can be used in any application and can increase the autonomous ability of a UAS, since most real life applications are performed in unknown environments. In [4], a micro air vehicle equipped with four cameras is using onboard computer vision combined with inertial measurement information in order to navigate autonomously in space while being able to avoid obstacles. A similar research has been introduced in [5], an onboard processing using stereo vision is done on a flapping wing micro air vehicle to avoid obstacles in generic environments. Automation of UAS aided by computer vision algorithms does not stop there. In [6] a technique for automatic vision-based take-off, terrain following and landing has been demonstrated, while in [7] a similar work has been presented. In [7] a UAV is performing automated take-off and landing on a moving platform using infrared cameras without the need of communication between the UAS and the platform. A similar work has been done in [PAPER] [8] where a color detection algorithm is performed to guide a UAV for landing in a moving vehicle. Another interesting research involving aerial vision processing is introduced in [9] where the UAS is automated with the support of computer vision algorithms to inspect and identify plantation failure. As can be seen, the popularity and demand in commercial low priced UAS keeps increasing. However, at the moment there are limitations in the technology available that make the task of autonomous UAS even more challenging. Vision is a fundamental part of a fully autonomous UAS. Nevertheless, being able to perform real-time onboard video processing for autonomous purposes while keeping the overall weight and power consumption to minimum but with a high processing ability makes the task really complicated and hard. In [10], this problem has been investigated by introducing alternative methods for onboard vision such as FPGAs and ASICs that have shown encouraging results. Another interesting research has been presented in [11]. A framework for choosing the best possible embedded platform for UAS on-board image processing has been introduced by taking into consideration parameters such as algorithm complexity, UAS velocity, power consumption and flight time.

III. FRAMEWORK ANALYSIS

In order to perform accurate visual analysis (i.e., identifying all objects/areas of interest correctly without false positive/negative detections) a sufficient amount of information must be collected (i.e. the object resolution must be sufficient). Johnson's Criteria defines the number of picture elements (pixels) required to discriminate an object on an image based on three main levels with more than 50% probability of success [12], [13]. These criteria have been drafted with a human operator in mind and for infrared image intensifiers and as such may not be precise for automated visual analysis tools. However, the same principles apply for other camera types and analysis options and thus can be used to build a basic understanding on the camera sensor features. According to Johnson's criteria, the three main levels of discrimination are as follows *a) detection*, *b) recognition* and *c) identification*. As suggested by the criteria, detection means that an object is present, and this is achieved when the object of interest has approximately 6 picture elements (pixels). For recognition, the type of the object should be distinguished; for example, whether the object is a car or human and it is achieved when the object has at least 80 pixels. The third level of discrimination is identification in which someone is able to identify the object, for instance it is able to distinguish whether a human is male or female. The identification level can be performed when the object has around 250 picture elements. *Figure 1* presents the three level of discrimination according to the picture element values.

For a given camera sensor, distance of the object from the camera will determine the discrimination level. As the object gets closer to the camera, more information about the object is visible and hence it covers more picture elements in the imagery taken from the sensor. The human brain is capable of effortlessly discriminating and classifying objects from among tens of thousands of possibilities almost instantly despite the tremendous variation in appearance that each object produces such as shape, color and texture [14], [15], [16]. However, despite the fact that Johnson's criteria have been introduced for humans, the concept can be applied in computer vision algorithms for object detection, thus, a minimum resolution depending on the object of interest, is required to perform detection (and identification).

Object detection involves a sliding window scanning the input image, in order to find objects of interest in various sizes. This search is usually done by extracting smaller regions from the frame, called search windows, of $m \times n$ pixels, which are

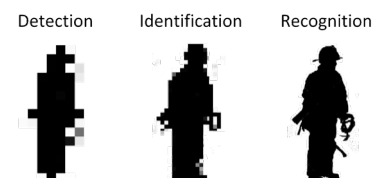


Fig. 1. Johnsons Criteria discrimination levels

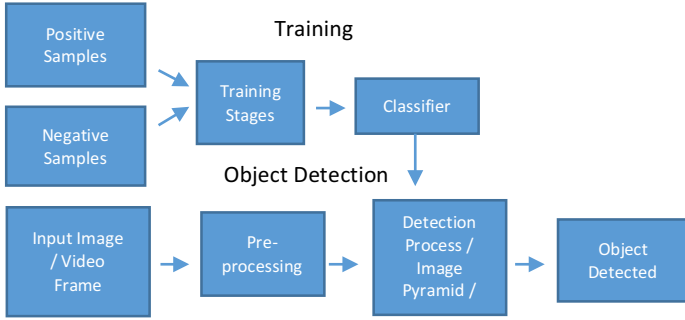


Fig. 2. Classifier training and object detection procedure

processed by a classification algorithm to determine if they contain the object of interest or not. To account for the variable sizes in objects, given the typically fixed size of the search window, an object detection system typically downscales the input image in steps (through a process known as image pyramid generation), effectively reducing the size of the object of interest, and re-examines the image, until the downscaled image is equal to the size of the search window. The classifier is being trained using a dataset consisting of positives and negatives samples in order to extract features of the object that has to be learnt. The dataset image samples used for the training, both negatives and positives, should have the same dimensions. The dimensions of the images can be used to calculate the recognition pixel value for a specific object detector. *Figure 2* illustrates the procedure of classifier training and object detection. The main problem of aerial object detection is altitude. Since the UAS altitude is inverse proportional to the object of interest size, the camera field of view geometry should be examined to adjust the object detection algorithm parameters accordingly.

Figure 3 illustrates a diagram of the geometry of the UAS and camera field of view. As shown in the figure, r is the actual metric distance of the field of view on the ground, h is the above ground level (AGL) altitude of the UAS and ϕ is the field of view angle as specified by the camera specifications. Most of the times, the field of view angles between horizontal and vertical axis along the ground are not the equal and hence

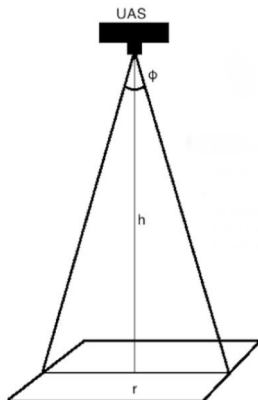


Fig. 3. UAS field of view geometry

the field of view ground projection is rectangle. By applying simple trigonometry identities on the geometry of the field of view as shown in *Figure 3*; an equation relating UAS altitude h , camera field of view angle ϕ and corresponding ground metric distance r can be derived as shown in (1).

$$h = \frac{r}{2 \tan\left(\frac{\phi}{2}\right)} \quad (1)$$

In order to introduce detection characteristics into (1), it is required to combine camera specifications and detector capabilities. Such parameters should include camera resolution, detector's training sample dimensions and desired object characteristics such as area. All these parameters can be related as shown in (2), where h_d (meters) is the desired/maximum UAS altitude that a recognition of a specific object can be done, \hat{p} (pixels) is the camera resolution either horizontally or vertically, $\hat{\phi}$ (degrees) is the corresponding camera field of view angle according to the selection of camera resolution, obj (square meters) is the top down object of interest and rec (square pixels) is the total number of pixels of a single training sample image used to train the object detector. Object of interest area obj , is the only parameter that is not given but can be estimated. Usually objects have some variation in dimensions such as vehicles where there is a range of values that can be as the average vehicle area. Some vehicles for instance are wider and longer giving a relatively high top down area while others are narrower and shorter giving relatively low area. However, (2) provides an estimation value of the desired altitude and hence, by using empirical knowledge and information someone can easily decide a reasonable estimation of the area for a specific object.

$$h_d \leq \frac{\hat{p}}{2 \tan\left(\frac{\hat{\phi}}{2}\right)} \times \sqrt{\frac{obj}{rec}} \quad (2)$$

As can be seen from (2), the resolution of the camera is proportional to the maximum recognition altitude. This is due to the fact that at a fixed field of view angle, higher resolution cameras will provide more information, allowing the object of interest to be recognized from increased altitudes. The maximum altitude is also proportional to the square root of the area of the object of interest. Obviously, as the object of interest area is getting larger it can be recognized from higher altitudes. On the other side, field of view angle is inverse proportional to the maximum UAS altitude. This happens because as the field of view is getting larger and the resolution of the camera is kept constant, the information captured from the camera is spread along the larger field of view. This phenomenon decreases the probability of an object being recognized. Equation 2, provides information about the capabilities of a specific aerial object detection algorithm in terms of altitude. By estimating the maximum altitude that an object detector is capable to perform effectively for a specific object using aerial images or video taken from a specific

camera sensor, provides the initial step of the evaluation framework. Consequently, the range of altitudes up to h_d meters will be called effective altitude range. The images or frames obtained from the camera sensor have a fixed resolution and field of view. Therefore, to maintain the accuracy of the algorithm along the range of effective altitudes, an image processing technique called image pyramid should be implemented in order to downscale images according to the altitude. This can be achieved by solving (2) in terms of image resolution \hat{p} as shown in (3). It is important to notice that the field of view angle used in (3), will give the corresponding downscaled image resolution. Hence, to find the other image resolution, the aspect ratio of the initial image has to be kept constant.

$$\hat{p} = 2h \tan\left(\frac{\phi}{2}\right) \sqrt{\frac{rec}{obj}} \quad (3)$$

As can be seen from (3), altitude and image resolution are proportional, meaning that as the altitude is decreased the corresponding ideal resolution is decreased too. By downscaling an image there is a loss of information meaning that the processing speed required is decreased. Consequently, object detection processing at lower altitudes can be performed faster. At real time object detection, the UAS has the capability to travel at higher velocities in lower altitudes while keeping the processed frame rate high. Nevertheless, field of view ground distance at lower altitudes is minor and it will require more travel distance in order to cover a pre-specified search area. On the other hand, higher altitudes, are restricted in lower velocities due to the high processing time required for each image, but the advantage of large field of view ground distance gives the opportunity to cover a pre-specified search area with less travel distance. All these parameters will be discussed to find the optimal solution. For real time object detection, it is important to maintain a balance between UAS velocity and CPU processing speed. For instance, if both UAS velocity and processing speed required to process a single frame are relatively high, then the processed frame rate will not be able to cope up with the UAS velocity. This phenomenon will lead to extreme loss of information since the UAS will be able to travel a relatively high distance while the detector will not be able to cover effectively this distance. *Figure 4*, illustrates a diagram of the movement of the UAS with velocity U (meters per second) travelled a distance d (meters per frame) in some time t (seconds per frame). The purpose is to find a relationship which will give the maximum velocity that the UAS will be able to travel at a specific altitude, bearing in mind the processing speed required for each single frame in order to maintain a smooth frame rate without the presence of lag and loss of information. After the maximum velocity is estimated, it is possible to use power consumption formulas in terms of aerodynamic forces using this velocity value to estimate the overall flight time at each altitude. By using the overall flight

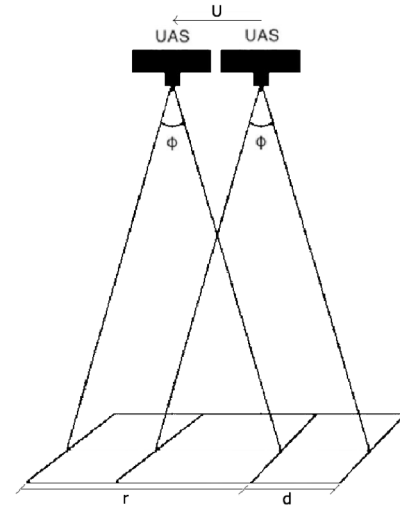


Fig. 4. The motion of UAS between two adjacent processed frames time estimation, a holistic performance evaluation can be achieved.

Equation 4, is the fundamental equation of motion at constant speed, where will be used to describe the motion illustrated in *Figure 4*.

$$U = \frac{d}{t} \quad (4)$$

To find the maximum velocity in each altitude as described earlier, it is required to introduce altitude in the travel distance d and a parameter that will represent the smoothness of the processed frame rate. It is important to state that, it is required to find an optimal travel distance per processed frame at each altitude. This will give an estimation of the velocity required for traveling to counterbalance the frame processing speed while maintaining a smooth processed frame rate. The travel distance is directly related to the altitude. However, to notice the change of travel distance at higher altitudes the travel distance should be larger compared to that in lower altitudes. This indicates that there is a ratio which relates the change in travel distance with the altitude. This ratio c (1/ frame) will be called distance ratio. Equation 5, relates the distance ratio c with the travelled distance d and field of view ground distance r . As can be seen from (5), distance ratio can be translated as the percentage of travel distance in terms of field of view ground distance.

$$c = \frac{d}{r} \quad (5)$$

Field of view ground distance can be directly related with altitude. Using simple trigonometry equations combined with (4) and (5), an equation relating maximum speed U , distance ratio c , altitude h , field of view angle ϕ and processing time required for a single frame t can be derived as shown in (6).

$$U = \frac{2ch \tan\left(\frac{\varphi}{2}\right)}{t} \quad (6)$$

Distance ratio values range from $0 < c < 1$. Normally the value should be close to zero, meaning that for a single frame processed, the travelled distance is minimal ensuring smooth processed frame rate that will be ideal for real time processing.

Since the maximum velocity in each altitude has been estimated, it is possible to introduce power consumption formulas that consider aerodynamic forces to estimate the overall flight time at each altitude using the corresponding velocity. Equation 7 and 8, shows the power produced by the propeller P_P (W) and the torque required to spin the propeller τ (Nm) respectively. Where T_T (N) is the total thrust produced by the propellers, n is the total number of motors/propellers, A (m^2) is the blade area, ρ ($= 1.225 \text{ Kg/m}^3$) is the air density, C_{DP} is the coefficient of drag of a single propeller and D_P (m) is the diameter of the propeller.

$$P_P = \frac{T_T}{n} \sqrt{\frac{T_T}{2n\rho A}} \quad (7)$$

$$\tau = \sqrt[3]{2\pi C_{DP} \rho P_P^2 D_P^5} \quad (8)$$

By combining (7) and (8), a relationship relating the required torque with the total thrust can be derived, as shown in (9).

$$\tau = \frac{T_T}{n} \sqrt[3]{\frac{\pi C_{DP} D_P^5}{A}} \quad (9)$$

The current drawn by a DC motor is related with torque as shown in (10). Where I_P (A) is the current drawn by a single motor and K_T (Vs) is the motor torque constant. Usually torque constant is not given and has to be replaced with a known parameter. This parameter should be motor constant K_V (rad/sV) and is related with torque constant with the relation presented in (11). Motor constant is given in the motor specifications. Combining (10) and (11), the current can be related with the motor constant and torque as in (12).

$$I_P = \frac{\tau}{K_T} \quad (10)$$

$$K_T = \frac{60}{2\pi K_V} \quad (11)$$

$$I_P = \frac{\tau \pi K_V}{30} \quad (12)$$

Substituting (9) to (12), a relationship that combines total thrust, motor and propeller specifications, with the current drawn can be derived as shown in (13). Where the motor's efficiency can be expressed as η .

$$I_P = \frac{T_T \pi K_V D_P}{30 n \eta} \sqrt[3]{4 C_{DP}} \quad (13)$$

The ultimate purpose is to introduce flight time in the equations. This can be achieved by using the current draw together with the battery capacity. However, I_P is the current drawn by a single motor. Equation 14, shows the total current drawn I_T (mA) by all motors.

$$I_T = 1000 n I_P \quad (14)$$

As long as the total current drawn is calculated, the total UAS flight time t_{EST} (*minutes*) can be found by (15). Where C_{BAT} (mAh) is the capacity of the battery that is used to power the UASs motors.

$$t_{EST} = \frac{60 C_{BAT}}{I_T} \quad (15)$$

Combining (13), (14) and (15) a final expression for total UAS flight time can be derived as shown in (16). Equation 16, relates the total flight time with the total thrust produced by the propellers.

$$t_{EST} = \frac{1.8 \eta C_{BAT}}{T_T \pi K_V D_P \sqrt[3]{4 C_{DP}}} \quad (16)$$

At this point, it is necessary to analyze total thrust. Total thrust corresponds to the aerodynamic force produced by propellers. However, to find total thrust it is important to realize the forces acting on a UAS. For the purposes of this paper, the UAS will be considered to be quadcopter. For simplicity purposes, it will be assumed that the UAS will either hover or travelling at constant velocity. Figure 5, illustrates the free body diagrams for both states hovering (see Figure 5a) and travelling at constant speed (see Figure 5b). While the UAS is hovering, the resultant thrust is acting on the center of mass with an upwards direction. In this situation Newton's Third Law is applied and hence, the weight force W (N) has an equal magnitude but opposite direction from total thrust. In the case where the UAS is travelling at a constant speed, the forces acting on it are becoming more complicated. During the constant speed movement, the backward propellers are both required to produce the same amount of thrust resulting to a

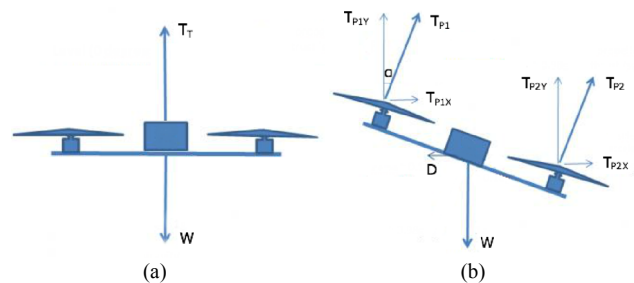


Fig. 5. UAS free body diagram in (a) hover state and (b) travelling at constant speed

total backward thrust, T_{P1} (N). This applies to the forward pair of propellers too which they produce a resultant forward thrust, T_{P2} (N). However, the total backward thrust and total forward thrust are not necessarily equal, but the total thrust is the sum of these two components. In order to perform thrust analysis in the UAS while travelling in constant velocity, it is required to find the vertical and horizontal components of both backward and forward thrusts as shown in *Figure 5b*. The total thrust component in horizontal dimension is equal to the drag force D (N), while the total thrust in vertical dimension is equal to weight. The drag force can be expressed as in (17), where C_D is the coefficient of drag of the UAS and S (m^2) is the frontal surface area of the UAS. The weight force can be expressed as in (18), where m (kg) is the total mass of the UAS and $g = 9.81$ (m/s^2) is the gravitational acceleration.

$$D = \frac{1}{2} \rho U^2 C_D S \quad (17)$$

$$W = mg \quad (18)$$

By performing simple trigonometry using the angle of attack α (degrees) of the UAS, it is possible to find an expression that will relate the total thrust produced by the propellers with the UAS velocity. This expression is demonstrated in (19).

$$T_T = T_{P1} + T_{P2} = \sqrt{m^2 g^2 + \left(\frac{1}{2} \rho U^2 C_D S\right)^2} \quad (19)$$

As can be seen from (19) if the velocity U is equal to zero, which means the UAS is hovering, the total thrust becomes equal to weight verifying the Newton's Third Law.

At this stage, all the parameters discussed can be combined. First of all, the processing speed required to process a frame at each altitude using a specific object detection algorithm has to be found. Then, the UAS traveling velocity can be calculated. By estimating the UAS velocity with (6) at a specific altitude, it is possible to find the total thrust required to travel at that velocity by using (19). As long as the total thrust is known for each velocity along the range of effective altitudes, the total flight time for each altitude can be estimated as shown in (16). To evaluate the performance of the detection algorithm in terms of total flight time and altitude, it is necessary to find how much search area can be covered while flying at different altitudes bearing in mind all these parameters.

IV. EXPERIMENTAL ANALYSIS

For the experimental analysis the equipment shown in Table 1 will be used. Additionally, the table illustrates important specifications or parameters that are related with each equipment. The next step, is to follow the procedure as explained in section III.

TABLE 1

| UAS | DJI Matrice 100 |
|------------------------------|------------------------------|
| Max Velocity | 22 m/s |
| MTOW | 3.4 Kg |
| C_{BAT} (TB47D) | 4500 mAh (each) |
| Propeller Diameter | 0.33 m |
| Coefficient of drag C_D | Approx. 0.8 |
| Frontal Surface Area S | Approx. 0.12 m^2 |
| Camera Sensor | X3 - Zenmuse |
| Resolution | 4000 x 2250 |
| Field Of View Angle - ϕ | 87.15 degrees |
| Processing Platform I | High-End Platform Early 2015 |
| CPU | Intel I7 3.1GHz |
| Processing Platform II | Nvidia Jetson TK1 |
| CPU | ARM - Cortex A15 2.3GHz |
| Weight | 0.241 kg |
| Processing Platform III | Raspberry Pi 2 |
| CPU | ARM - Cortex A7 900MHz |
| Weight | 0.08 kg |

For the purpose of this investigation, an object detection algorithm trained to detect vehicles has been used. Just for reference, the algorithm has been trained using Histogram of Oriented Gradients (HOG) and approximately 4000 positive samples and around 50000 negative samples have been used, each of them had a size of 40 x 20 pixels. The first step on the procedure is to find the maximum altitude that the UAS will be able to travel according to the camera specifications, while the detector will be able to perform effectively. Although this

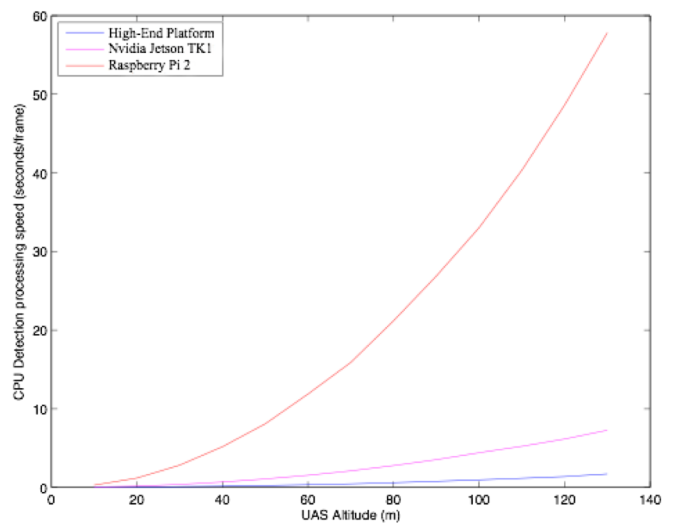


Fig. 6. Shows the processing speed required to process a single frame, at each altitude

altitude has been found to be approximately 200 meters high, for safety reasons and to comply with airworthiness regulations we reached a maximum altitude of 130 meters. So the effective altitude range for our experiment was 0-130 meters. Starting from an altitude of 10 meters increasing the altitude with a step of 10 meters, top down images have been taken all the way up to 130 meters. A total of 13 images have been collected and each of them has been resized accordingly using (3). Each image has been processed through the vehicle detector using the three available processing platforms as shown in Table 1. The processing speed required by each processing platform for each frame at each altitude has been plotted as shown in *Figure 6*.

As shown from *Figure 6*, it is obvious that generally for all processing platforms as the altitude is increasing, the processing speed required to process a single frame is increasing exponentially. This is expected since for higher altitudes the resolution of the image required to perform object detection is higher, meaning that more information have to be processed demanding more processing time. When designing an aerial object detection algorithm, it is important to bear in mind that altitude indeed affects the performance of the algorithm, especially if the processing has to be real time. The following step is to calculate the UAS maximum velocity at each altitude for each processing platform according their capability in terms of frame processing speed using (6). If the processing speed required to process a single frame is very low, the velocity restriction may be too high and practically impossible to achieve. Therefore, it is essential to consider the UAS maximum velocity, according to manufacturer specifications, as the velocity upper limit. Regarding the distance ratio c , it is decided to use all the range between $0 < c < 1$, in order to have a spherical overview of its impact. The next phase is to estimate the total thrust required to achieve each velocity using (19). At this point it is possible to estimate the total flight time of the UAS at each altitude using (16). However, using total flight time is not able to evaluate the performance of an object detection algorithm. Nevertheless, using flight time, corresponding UAS velocity

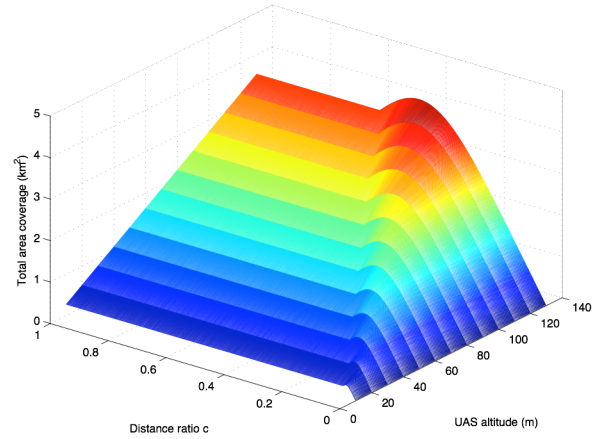


Fig. 8. Shows the relationship between the range of altitudes and corresponding total area coverage for all values of distance ratio c , according to Nvidia Jetson TK1 processing performance

and field of view ground distance, it is possible to estimate the total area that the UAS will be able to cover. *Figures 7,8 and 9* present a 3-Dimensional plot of the UAS altitude, distance ratio and total area coverage for High-End Platform, Nvidia Jetson TK1 and Raspberry Pi 2 respectively. Obviously High-End Platform is not able to perform onboard processing and hence, to maintain the comparison at the same standards these three plots consider that the processing is performing on the ground. As can be seen from *Figure 7*, where the performance of High-End Platform is illustrated, there is a minimum distance ratio value at each altitude at which any greater values distance ratio the total area coverage remains the same. This happens because as the distance ratio is getting larger, the overlay between adjacent processed frames is getting lower. This means that the distance between adjacent processed frames is getting larger allowing the processing platform to have additional available time to process a frame while the UAS is travelling to cover the larger distance towards the next frame. Since High-End Platform has high processing capabilities, this gives the opportunity to the UAS to travel at high speeds and theoretically exceeding the UAS

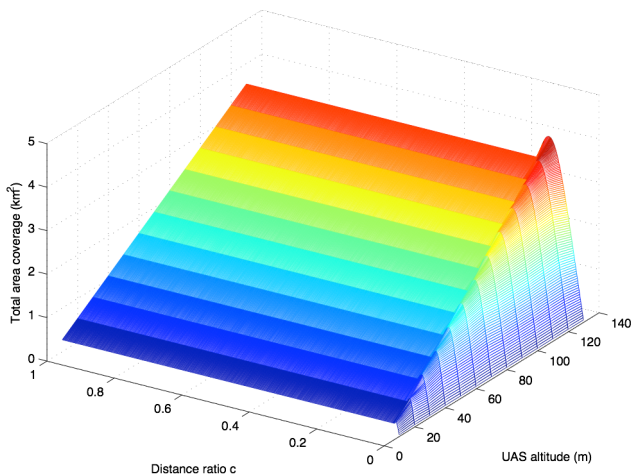


Fig. 7. Shows the relationship between the range of altitudes and corresponding total area coverage for all values of distance ratio c , according to High-End Platform processing performance

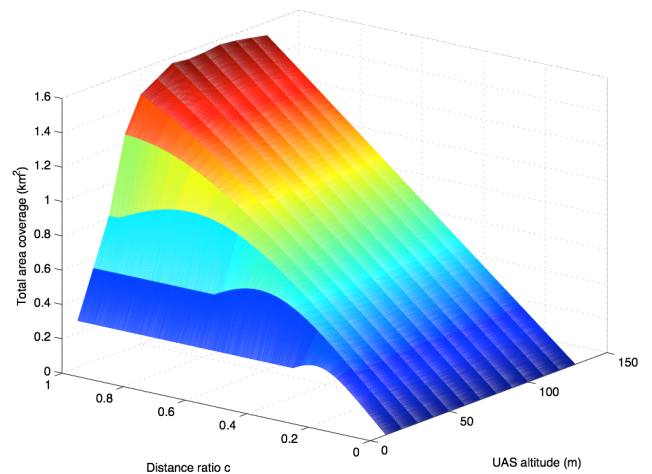


Fig. 9. Shows the relationship between the range of altitudes and corresponding total area coverage for all values of distance ratio c , according to Raspberry Pi 2 processing performance

maximum specifications velocity for these distance ratio values. In practice this is not possible and hence the UAS is travelling at its maximum allowed speed. Consequently, the total flight time remains the same at each altitude for all these values of distance ratio providing the same total area coverage at each altitude. This can also be noticed in *Figure 8*, where the same applies to Nvidia Jetson TK1. However, High-End Platform is more powerful in terms of performance capabilities compared to Nvidia Jetson TK1, therefore, comparing the corresponding altitudes in both *Figures 7 and 8*, it can be noticed that the minimum distance ratio value at which the total area coverage remains unchanged, is lower for High-End Platform compared to Nvidia Jetson TK1. As can be seen from *Figure 9*, the same applies to Raspberry Pi 2 but in much lower scale. Due to the fact that Raspberry Pi 2 has significantly lower performance capabilities compared to the other two platforms, only at altitudes of 10-20 meters there exist a minimum distance ratio value at which the total area coverage remains unchanged. Comparing *Figures 7,8 and 9* it can be clearly seen that there is a distance ratio value at each altitude that gives the maximum total area coverage. It can also be noticed that these optimal distance ratio values form a linear relationship with altitude. Additionally, the gradient of this linear relationship is getting lower as the processing platform is more powerful. As explained earlier in Section III, the distance ratio value is expected to be close to zero to ensure a smooth frame rate and being ideal for real time processing. This means that state of the art processing platforms will have minimal gradient between optimal distance ratio and altitude as shown in *Figure 7*. Since High-End Platform has high performance capabilities, the optimal distance ratio values are all close to zero (minimal gradient), meaning that for all the effective altitude range High-End Platform is more capable in performing real time object detection compared to the other two platforms. A hardware designer that is looking forward to design a processing platform ideal for aerial real time object detection, can determine a maximum distance ratio value for real time object detection and then by using this framework can evaluate the performance of the hardware and software in terms of flying altitude for optimized area coverage.

As said earlier, the processing in this experiment has been done on the ground, nevertheless, if the processing is being held onboard then the overall mass of the UAS will be increased due to extra processing platform weight and hence the power consumption will have an increase. These parameters will not affect the shape of the plots. The only impact will be the total area coverage that will be decreased by some percent since the overall time will be decreased as a result of the increased weight and power consumption. Since Raspberry Pi 2 is low weight, low power consumption processing platform will have minimal impact on the area coverage compared to that of Nvidia Jetson TK1.

V. CONCLUSIONS

This paper demonstrates a holistic performance evaluation framework for aerial object detection platforms and algorithms. It has been shown how various parameters such as altitude, camera specifications and aerodynamic forces affect the performance of an object detection algorithm. A hardware or software designer aiming to develop a new hardware or software specifically designed for real time object detection onboard a UAS should consider all these factors. This framework can be used as a guideline to evaluate the performance of the processing platform and object detection algorithms. Most real life applications that could benefit from an autonomous UAS require aerial monitoring and surveillance. Knowing in advance how all these parameters are related to each other and being able to find the optimal area coverage while maintaining the ability of real time processing is vital for designing the best possible solution.

VI. REFERENCES

- [1] H. P, "Framework for Vision-Based Power Line Inspection with an UAV," Swiss Federal Institute of Technology Zurich, Zurich, 2012.
- [2] Hersha J, Johnson M, Meredith I, Pittsley D, Tameesh F and Wilson C, "Drone Monitoring of Power Lines," Michigan State University, Michigan, 2015.
- [3] Doherty P and Rudol P, "A UAV Search and Rescue Scenario with Human Body Detection and Geolocalization," in *AI 2007: Advances in Artificial Intelligence*, Gold Coast, Australia, 2007.
- [4] Meier L, Tanskanen P, Fraundorfer F and Pollefeys M, "PIXHAWK: A System for Autonomous Flight using Onboard Computer Vision," in *IEEE International Conference on Robotics and Automation*, Shanghai, 2011.
- [5] De Wagter C, Tijmons S, Remes B.D.W. and D. C. G.C.H.E., "Autonomous Flight of a 20-gram Flapping Wing MAV with a 4-gram Onboard Stereo Vision System," in *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014.
- [6] Ruffier F and Franceschini N, "Visually guided Micro-Aerial Vehicle: automatic take off, terrain following, landing and wind reaction*," in *IEEE International Conference on Robotics and Automation*, New Orleans, LA, 2004.
- [7] Wenzel K.E., Masselli A and Z. A., "Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 221-238, 2011.
- [8] L. Hanseob, J. Seokwoo and H. S. David, "Vision-based UAV Landing on the Moving Vehicle," in *International Conference on Unmanned Aircraft Systems*, Arlington, 2016.
- [9] da Silva J.F., Brito A. V. and d. M. H. N., "An Embedded System for Aerial Image Processing from Unmanned

- Aerial Vehicles," in *IEEE Brazilian Symposium on Computing Systems Engineering (SBESC)*, Brazil, 2015.
- [10] Ehsan S and K. D. McDonald-Maier, "On-Board Vision Processing For Small UAVs: Time to Rethink Strategy," in *NASA/ESA Conference on Adaptive Hardware and Systems*, San Francisco, 2009.
- [11] Hulens D, Verbeke J and G. T, "Choosing the Best Embedded Processing Platform for On-Board UAV Image Processing," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics*, Berlin, 2015.
- [12] "Johnson's Criteria for Pixel Resolution," 2016. [Online]. Available: <http://www.trec.com/johnsoncriteria.htm>.
- [13] N. S. Kopeika, *A system engineering approach to imaging*, SPIE Optical Engineering Press, 1998.
- [14] I. Biederman, "Biederman, I. Recognition-By-Components: A Theory Of Human Image Understanding.," *Psychol.Rev.*, 1987, pp. 115-147.
- [15] M. Potter, Potter, M.C. Short-Term Conceptual Memory For Pictures., *J. Exp. Psychol. Hum. Learn.*, 1976.
- [16] N. Logothetis and D. Sheinberg, in *Visual object recognition*, *Annu. Rev. Neurosci.*, 19, 1996, pp. 577-621.
- [17] R. Austin, *Unmanned air vehicles*, Chichester, West Sussex, UK: Wiley, 2010.
- [18] FAA, "Unmanned Aircraft Systems (UAS) Frequently Asked Questions/Help," 2016. [Online]. Available: <https://www.faa.gov/uas/faqs/>. [Accessed 10 4 2016].