A New Approach for Vehicles Detection in Low Altitude Aerial Images based on Edge Density

Antonio J. R. Neves, Manuel Camarneiro and Lucas Cozinheiro

DETI/IEETA, University of Aveiro,
3810-193 Aveiro, Portugal
Email: {an, mcamarneiro, lucas.cozinheiro}@ua.pt

***Abstract*—Drones and Computer Vision are both important research topics nowadays. Combining these two technologies enables having different perspectives of a scene and capturing them with digital cameras onboard drones. Incorporating onboard processing of the acquired images might provide extremely useful information from different scenarios, allowing the drones to perform autonomously with applications in several contexts. In this paper, we present algorithms to process images captured by drones over parking lots in order to detect parked vehicles and further estimate occupancy rates or cars parked in a wrong place. As far as we know, low altitude image processing is still an open problem in the computer vision community. Experimental tests with the developed algorithms were performed in different parking lots across the University of Aveiro Campus under different lighting conditions to check their accuracy and processing requirements. Detailed experimental results are presented in this paper and an image database was produced in order to allow future experiments by other researchers. The obtained results presented in this paper demonstrate the effectiveness of the proposed approaches.**

***Keywords*–Drones; Image processing; object detection.**

I. INTRODUCTION

This manuscript is an extended version of the original paper presented at the Second International Conference on Advances in Signal, Image and Video Processing, SIGNAL 2017 [1]. This extended version provides a deep overview about the proposed algorithms for car detection on low altitude images and more experimental results using these algorithms.

Recent drones are equipped with digital cameras and are very prospective for a variety of commercial uses such as aerial photography, surveillance, etc. However, in order to deploy autonomous drones, it is necessary to include onboard smart computer vision systems and autopilot capabilities. In the application of aerial imaging, object detection and tracking are essential to capturing key objects in a scene. Object detection and tracking are classic problems in computer vision. However, there are more challenges with drones due to top-down view angles and real-time constraints. Additionally, a challenging problem is the strong weight and area constraint of embedded hardware that limits the drones to run computation intensive algorithms, such as deep learning, with limited hardware resource.

Solutions using drones and computer vision are not restricted to security. There is a huge number of possible areas where these devices might be useful [3] [5]. Although, only a few commercial drones are able to perform some basic image processing on the acquired images. There is still a long way to go through on scientific research about this technology combination. Lately, a few commercial solutions are available for applications in agriculture mainly used to monitor plants growth, watering levels and fruit maturation. Some prototypes are also being tested for save and rescue tasks or fast mail delivery.

Before the proliferation of drones, monitoring vehicles from aerial imagery was already possible, making use of pictures either taken from satellite or from manned aircrafts. For this kind of images there are several approaches regarding algorithms to detect and locate cars. This is often associated with high altitude or satellite imagery [4] [7]. Even though, as this project was intended to process images obtained in lower altitude flights (about 10 meters from the ground), most of the published work does not apply. Thus, the solution was to create algorithms from scratch for parking lots with three different types of pavement, given as inputs the altitude of the drone and the parking zone location in relation to the main road.

The experimental results presented on this work are based on images captured using a Cheerson CX-20 and a Parrot Bebop 2 [2] flying on a selected path over some of the University of Aveiro parking lots spread across the Campus as shown in Figure 1.

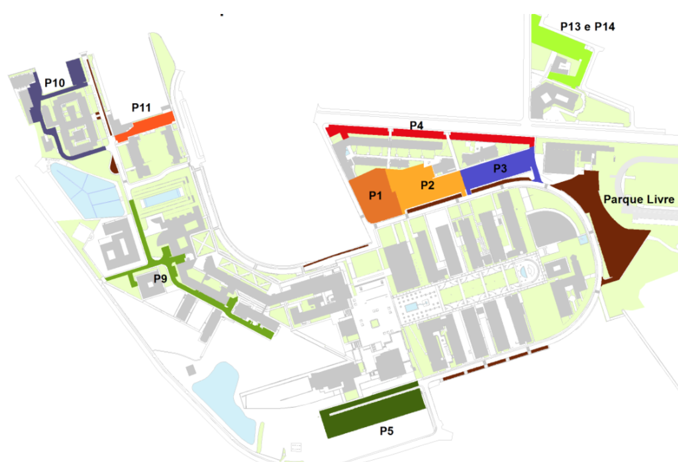

Figure 1. University of Aveiro Parking Lots.

Images were captured over some parking lots taking into consideration the type of pavement. Images were obtained either from pavements built exclusively on tar (Figure 2), on block pavement (Figure 3) and both (Figure 4). Algorithms were further developed to detect parked vehicles over each type of pavement identified before.

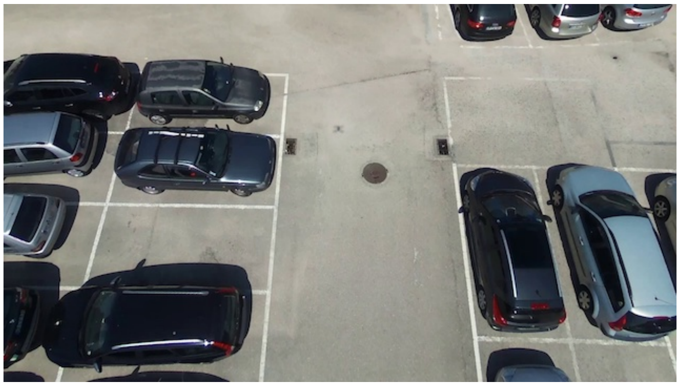

Figure 2. Tar Parking Lot.

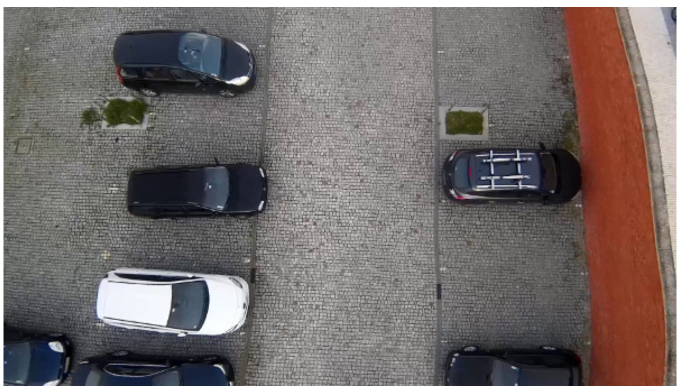

Figure 3. Block Pavement Parking Lot.

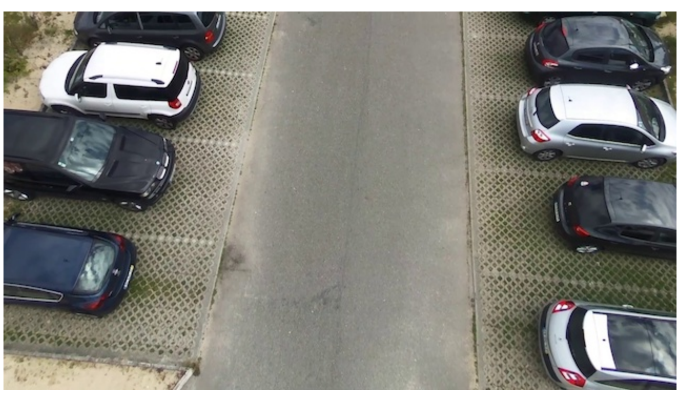

Figure 4. Mixed Pavement Parking Lot.

The algorithms were tested on an external computer used for development but were also adapted for further tests in single boards in order to determine the possibility of having image processing onboard while the drone moves over the parking lots. Without local processing, this solution would either depend on a fast and heavy data transfer link, for captured images to be processed in a remote server, or take a longer time gap since the drone is sent to capture images, returns and delivers data acquired on its base station for further processing and analysis.

In this paper we present experimental results showing the effectiveness of the proposed approach, both in terms of detection ratio, as well as in terms of processing time.

The paper is organised as follows: Sections II and III introduce respectively Drones and relevant related work. Sections IV to VI detail the proposed solution from an overview on Low Altitude Vehicle Detection, to our solution Pavement and Vehicle Detection. Section VII is reserved for experimental results, while finally, in Section VIII, we draw some conclusions.

II. Drones

Unmanned Aerial Vehicles (UAV) or commonly denominated Drones are aircrafts able to fly without a human pilot on board and can either be remote controlled or simply fly autonomously. UAV were firstly developed for military purposes but quickly became popular. By the end of the last century some people already managed to control small aircrafts remotely but brushless motors and new batteries brought some more efficiency to these models, allowing them to operate only on electric power and quickly lead to further research in this area.

Different designs came up alongside scaled planes and helicopters but multicopters stood out. Multicopters or Multirotors are in fact rotorcrafts with 2 or more motors, combining the features of a regular helicopter - vertical take-off and landing and hovering capability - with the stability obtained by the extra motors. The quadrotor or quadcopter soon appears as a stable model, as well as efficient and easy to control, which makes it the most usual UAV flying these days. It is composed by a X-shaped body with 2 opposite propellers rotating clockwise and 2 other rotating counterclockwise illustrated in Figure 5.

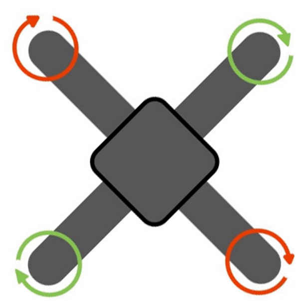

Figure 5. Quadcopter scheme.

A. *Used Drones - Cheerson*

Quanum Nova or Cheerson CX-20 is a commercial drone with an open-source ArduPilot Mega (APM) flight controller assembled in a fragile plastic frame presented in Figure 6.

Besides the customisation possibilities offered by the flight controller it also features a GPS module supporting path planning (pre programmed route from a smartphone) and a 2D stabilised gimbal for cameras. Stability is evident after some flying tests remotely controlled. Controlling range is also considerable. Nevertheless, a First Person View System(FPV)

Figure 6. Cheerson CX-20.

on board is missing - this shows up to be very important while flying for image acquisition. This drone also has some failsafe options and in case the pilot is not able to land, he can still touch two buttons making it return to the take off position. GPS accuracy is not disappointing, still it misses the target by 3 to 5 meters.

Quanum Nova is very versatile for aerial imagery solutions due to its stability, strength and design, which enables fixing and transporting a large diversity of action cameras and stabilisation gimbals onboard. Even though installing telemetry and activating pre defined paths is difficult and makes it harder to get it completely autonomous and able to repeat some footage over the same parking zones. Flight time goes up to 15 minutes but it is significantly reduced if using a camera and specially a stabilisation gimbal, which also compromises image acquisition over some big parking lots.

B. Used Drones - Bebop

The second UAV operated was Parrot Bebop2, which is a small sized quadcopter with a built-in panoramic camera (Figure 7).

Figure 7. Bebop 2.

It is controlled over Wi-Fi and does not require a specific remote. Any smartphone or tablet (having Parrot application installed) will be quickly transmitting all the instructions needed to determine the tasks of the Bebop. Range is not great but as it is possible to upload a well defined path, range is not a constraint. At a first look it seems small and fragile but it proved to be quite resistant to strong winds. It is also extremely easy to pilot since it takes off and lands automatically while keeping steady in case of no remote input.

In the smartphone or tablet it is possible to watch real time images obtained by the camera (FPV) and also have sensorial information such as altitude, speed or temperature. These features are crucial while obtaining images from parking lots since they enable visual feedback in real time and consequently allow the pilot to correct the position of the drone. The built-in camera is the only limitation of this drone when compared to Quanum Nova. Rotation is not available but Parrot software performs some lens and movement correction to obtain an image less distorted and almost perpendicular to the ground.

Camera and image processing onboard play an important role in the whole sensorial system since they are responsible for calculating drone speed. Taking images every 16 milliseconds Parrot software computes ground displacement and from it estimates drone speed. Despite having a battery similar to the ones used in Quanum Nova, flight times are almost doubled. Significant power losses caused by camera stabilisation drives (what happened in Quanum Nova) were eliminated and its light weight roughly contributes for a longer battery life.

III. Related work

Before the proliferation of drones, monitoring vehicles from aerial imagery was already possible, making use of pictures either taken from satellite or from manned aircraft. For this kind of images there are several approaches regarding CV algorithms to detect and extract cars position.

Jae-Young Choi and Young-Kyu Yang developed algorithms to detect vehicles in high altitude aerial images [4].

They started by finding small blobs which size should correspond to a car, using mean shift clustering algorithm. By choosing a radius interval, maximum and minimum car size, they could find regions of interest in the image.

After that, assuming the rectangular shape of the car, they evaluated the shape formed by the edges inside each blob.

Finally, merging blobs overlapped would eliminate vehicles counted more than once, happening when windshield and car color don not match.

Two images where this algorithm was applied are presented in Figure 8 alongside detection results.

Evaluation criteria	Left image	Right image
Number of detection	63	129
False alarms	5	11
False negative alarms	12	16
Detection rate (%)	84.0	88.9
Accuracy rate (%)	92.1	91.5

Figure 8. Choi Vehicle Detection Results [4].

Tuermer also published some work on algorithms for vehicle detection on aerial images obtained from regular piloted aircrafts [7]. Images are captured at a minimum altitude of 1000m with high resolution cameras. Still in the best cases each pixel represent a 15cm square on the ground.

Developed algorithm consisted of three main parts:

- Pre processing: Smoothing image and applying region growing technique on pixels with similar RGB values (color information). This separates interest zones to be further processed.
- Calculate histogram of Oriented Gradients (hOG) for each interest region created before.
- Decide if interest area is a vehicle using cascade classifiers.

After processing and evaluating some training image sets, algorithm was tested achieving about 80 percent of confidence on detected vehicles. In Figure 9 a sequence detection using Tuermer's solution is presented.

The two presented examples were chosen to illustrate some of the best techniques proposed for vehicle detection in high altitude images. Although there are several other approaches none of them are applicable to this project. No recent investigation in this area was found and all approaches assume high altitude images and ignore if the car is in a parking place or moving in the road.

In this paper the main objective is achieving fast image processing for low altitude images and aims to detect only parked vehicles, a new algorithm was written from scratch.

IV. Car Detection in Low Altitude Images

To evaluate parking lots capacity the first mandatory task is image acquisition. Assuming the drone is correctly positioned regarding the road, a frame should be captured and sent to the image processing unit. Once there, the image might need to be corrected in case of heavy distortion effects. After this, the algorithm should try to detect vehicles, compare them with others detected in previous images to check if they were already counted, and finally update the counter. This repetitive pipeline is presented in a circular graphic in Figure 10.

Vehicle detection is obviously a decisive part of software but a broad range of cars might appear in a parking lot. Features like color, size or shape may vary from one to another making it harder to create a global solution capable of detecting them all based only on these features. At the same time, it is necessary to ensure that detected objects are effectively vehicles, distinguishing them from similar objects that might appear.

Distinguishing pavement from other objects placed above the ground could be a possible solution. Even though, this sets up another challenge. Homogeneous gray concrete or tar might be easily discarded using a color filter threshold. Nevertheless, the same technique will not work in a park built in block pavement.

Taking into consideration all the challenges presented above, the global algorithm for car detection based on edge density analysis is divided into three main parts:

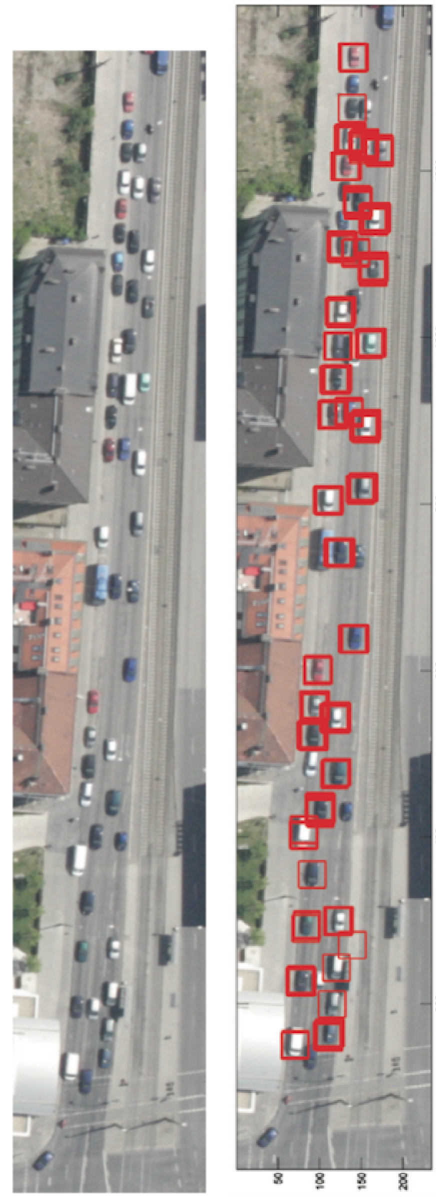

Figure 9. Tuermer Vehicle Detection Results [7].

- Road Detection - Different types of pavement will look completely different from an aerial perspective, but it is crucial to detect the central road of the parking lot in order to determine the parking places which are distributed on the sides of the road.
- Select Zones where vehicles are expected to be parked - After having the road segmented, it is possible to estimate its width, comparing it to the parking places size, thus confining the searching window.
- Vehicle Detection- Analyse features like color and edge density on the regions of interest in order to detect if there are parked vehicles there.

The most important modules of the proposed algorithm are represented in Figure 11.

V. PAVEMENT DETECTION

As stated before parking lots are built over different types of pavement which directly affects the chosen image process-

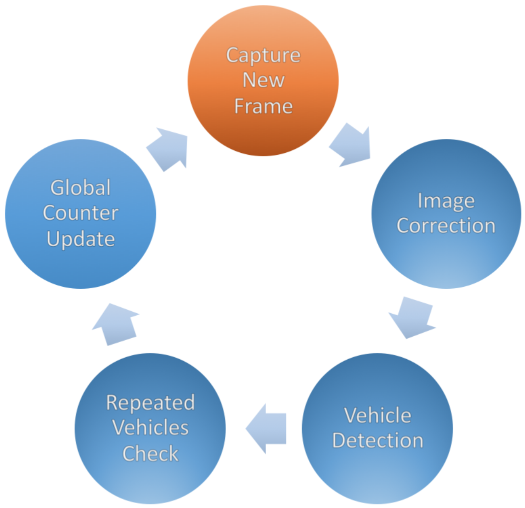

Figure 10. Global Solution Pipeline.

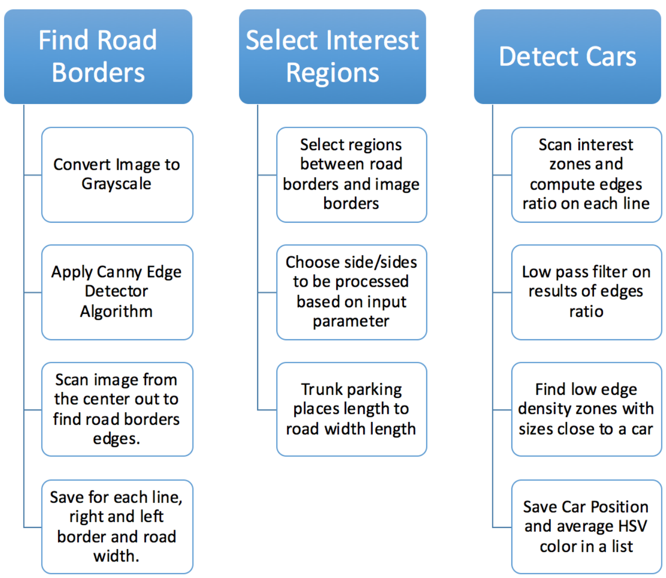

Figure 11. Main modules of the proposed Vehicle Detection Algorithm.

ing method. In order to turn the algorithm suitable for the three different types found in the University parking lots different approaches were used.

A. Block Pavement Detection

Canny Edge Detector algorithm [6] was used as a fast and optimized method to perform gradient computations and retrieve the most important edges for each acquired image. Figure 12 shows a fine mesh, which corresponds to the edges of each small block that composes the pavement. Cars, on the other hand, are found in zones of low edge concentration.

It would be possible to simply cluster regions with low edge density and compare their size to the expected car size (which estimation would depend on the altitude of the drone). Even though this would give space to detection errors, either by including more than one car in a single cluster or by analysing uninteresting zones in the surrounding areas. Some gardens or sidewalks, for instance, feature smooth surfaces making it harder to distinguish them from parked vehicles.

Figure 12. At the top, an image of a Block Pavement Parking. On the bottom the corresponding gradient Image.

The solution was finding the road borders to further estimate parking places position. After applying a color filter and Hough lines detector to locate the grids along the road, it is possible to establish the interest regions, on the left, right, or both sides of the road (Figures 13 and 14).

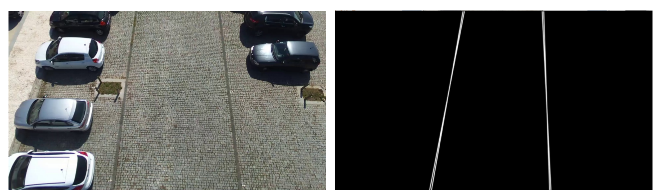

Figure 13. On the left, the image acquired by the drone, on the right the road borders detected.

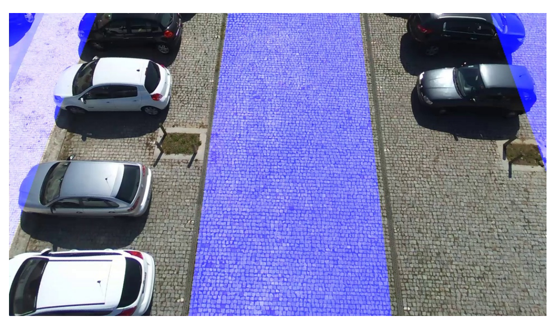

Figure 14. Road is removed as well as unwanted lateral zones. The length of the parking places is very close to access the width of the road, thus interest region is trunked as presented.

B. Tar Pavement Detection

Images obtained over tar pavement are smoother and lack of edges when compared to blocks pavement presented before. Despite that fact, it is still possible to reuse the logic from the last algorithm, detecting the road using the limit lines and trunking the interest regions.

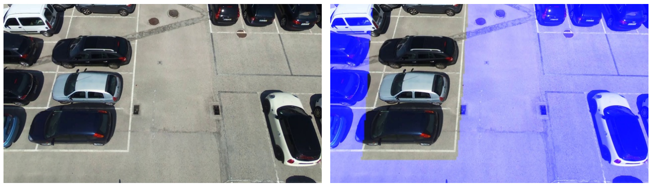

Figure 15. On the left, the image acquired by the drone, on the right the road borders detected.

As tar is homogeneous either in texture as in colour, checking if a low edge density zone is free or occupied can be done using color matching, making sure it is different from the tar found in the road (Figure 16).

Figure 16. On the left, the image acquired by the drone, on the right the vehicle detection on tar parkings.

C. Mixed Pavement

Most of the studied parking lots are built on both tar (used in the access road) and blocks with different configurations (used in parking places). The algorithm developed for this type of pavement slightly differs from the others, given the impossibility of detecting road based on color filters.

In this case, road limits are determined using the same notion of edge density. Tar zones are not expected to have high gradient values thus road might be easily detected. Further vehicle detection is performed as explained for the block pavement, as well as vehicle repetition check (see Figures 17 and 18).

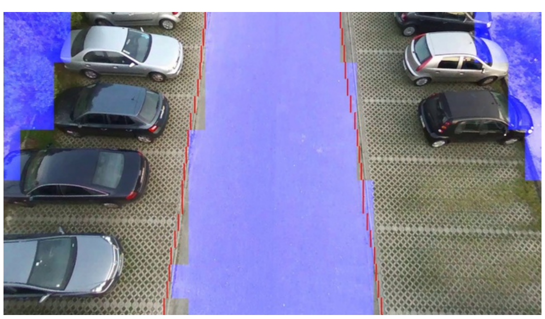

Figure 17. Road detection in mixed pavement parking lots. Raster scan window used to evaluate edge density is variable and affects processing times and road detection accuracy.

VI. VEHICLE DETECTION

Having the road segmented and a valid estimation of the regions corresponding to parking places, edge density analysis may now be performed for each one of these regions.

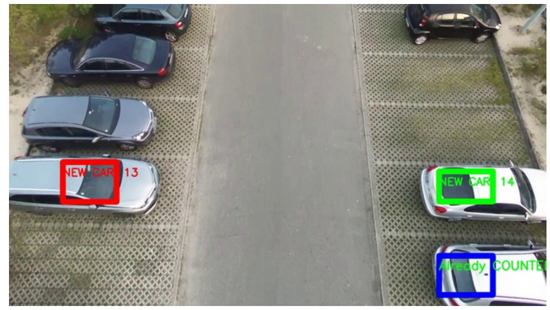

Figure 18. Vehicle detection in mixed pavement parking lots.

A. Edge Density Analysis

This analysis is done across the Y axis of the image (from the top to the bottom), evaluating the ratio of pixels which do not belong to edges. As vehicles often display some edges (dividing the doors from the roof for example) and considering them could be a reason for the algorithm to detect a division between two distinct cars. To eliminate this issue first step consisted in setting a minimum distance between cars, and the second step consists in reducing noise interference, visible in high frequency signal plotted. To achieve it a low pass filter is applied and the result is shown in Figure 4.21. Figure 19 was the picture used for the edge density analysis, where it is expected to find two significant low edge density peaks, corresponding to the cars on the right side of the image. Computed data is then plotted (Figure 20) and a lower pass filter is applied (Figure 21).

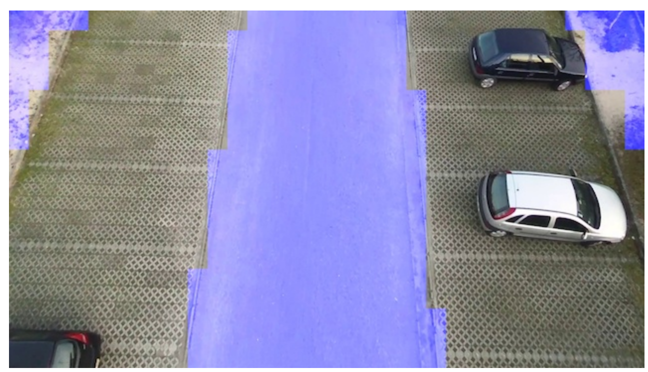

Figure 19. Sample image where edge density analysis is performed.

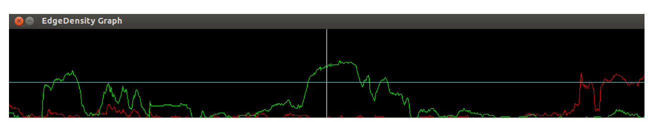

Figure 20. Green line peaks represent low edge concentration along the right region of interest, while red line represents the same analysis on the left side of the road.

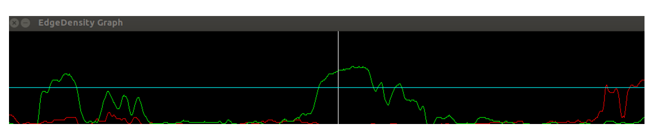

Figure 21. Edge density graph after application of a low pass filter. Noise reduction is noticeable.

The expected size of the vehicle is now compared with the size of detected stains according to their position on the image.

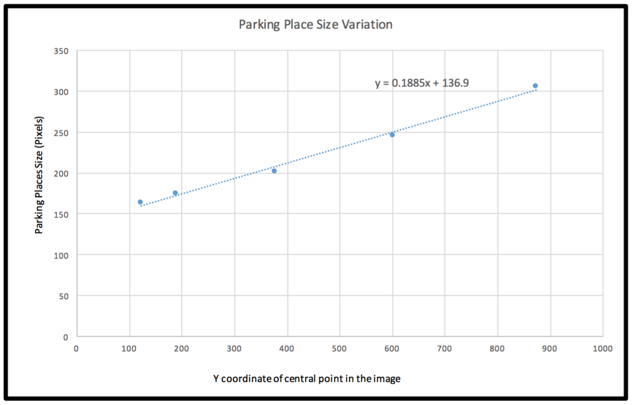

Figure 22. The area occupied by similar parked vehicles varies across the image with the distance from them to the camera.

Objects on the top (lower Y values) appear smaller due to the perspective introduced by the Drone used for image acquiring, still the vehicle size varies almost linearly. As the tilt angle of the camera is kept approximately constant throughout the flight, this linear variation is the same for all acquired images. This allows the computation of the expected vehicle size according to its location within the image. Even though, on the top of the image with such a small expected vehicle size, it might be difficult to distinguish and split two or more cars parked next to each other. To solve this issue the algorithm analyses only the lower half of the image where vehicle representation is more evident. In case of having a car right in the middle of the picture the algorithm keeps looking on the upper part of the image for the end of this low edge density region.

B. Repeated Vehicles Check

Despite the drone is moving with an almost constant speed, it is not possible to capture images without any overlay, meaning that the same vehicle might be present in more than one frame. To avoid double counting, it is required storing color, size and position features of vehicles detected in the last frame to compare them with the vehicles detected at the moment (Figure 23).

Figure 23. New and Repeated Vehicles Detected

VII. EXPERIMENTAL RESULTS

This section presents important values measured and analysed for the studied solutions. These are related to detection accuracy and processing times. Tests were applied on sample videos captured over different parking lots: two on tar pavement, two on block pavement and four on mixed pavement (block and tar).

All tests were performed using an Unix distribution (Ubuntu 14.04.3) installed on a computer with an Intel Core i5-3340M CPU @ 2.70GHz 4 processor with 4Gb RAM. Images captured were recorded as video and split into frames considering only one frame each half a second. A splitting tool was also developed to read each frame of the video and save images every 500 milliseconds in a specific directory previously defined.

Car detection accuracy is evaluated frame by frame comparing manual annotation of the number of cars depicted with detection boxes drawn by the algorithm.

It is crucial to choose a suitable edge detection method since these operations are performed every time a new image is processed. It is important to ensure some points regarding the chosen method:

- Detects low edge concentration over the road pavement (in case the road is made of tar)
- Creates high edge density zones over block pavement, contrasting with uniform surfaces on vehicles.
- Takes a short period of time to compute all the edges in an image.

Choosing the most suitable values enables accurate detection of homogeneous regions as shown in some examples presented in Figure 27.

It is now evident that Canny is an optimised edge detection method, possible to adapt to different situations by conveniently adjusting its parameters. It also features less processing requisites when compared to Sobel making it the best method and the one used for the rest of the algorithm tests.

Finding road limits composes a crucial step in the pipeline of the algorithm since this is performed in every park and is essencial to the location of interest zones. It is not relevant to have high accuracy in this procedure as the main goal is to eliminate the major region of the image representing road. It is not decisive to remove every single pixel from the road and preserve all other pixels, what is mandatory is the task to be quickly executed in every frame captured. Even though, road detection methods are distinct for different parking lots and different solutions should be developed for different pavement types.

On the other hand, vehicle detection based on low concentration of edges over the interest regions should be as accurate as possible and is applicable with only a few parameters adjustments to all studied parking lots. The following sections will detail results obtained for each type of parking lot studied.

Parking Lots in homogeneous kind of pavement are perhaps the most simple to deal with. Higher detection rates are then most likely to happen in P1 (Table I and Figure 25).

The block pavement revealed to be a difficult background to extract vehicles from. Despite edge density zones concept being easier to imagine in this situation, region segmentation

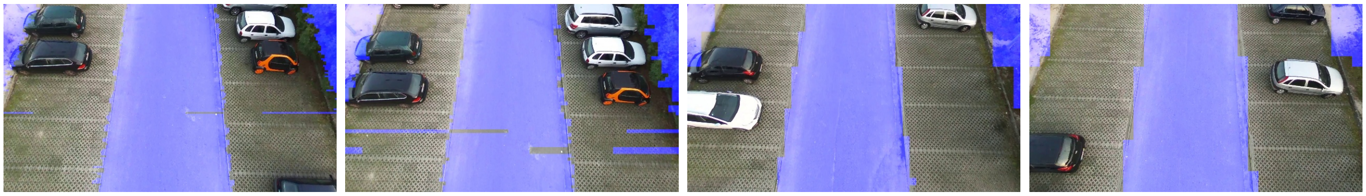

Figure 24. Sliding window influence on road detection.

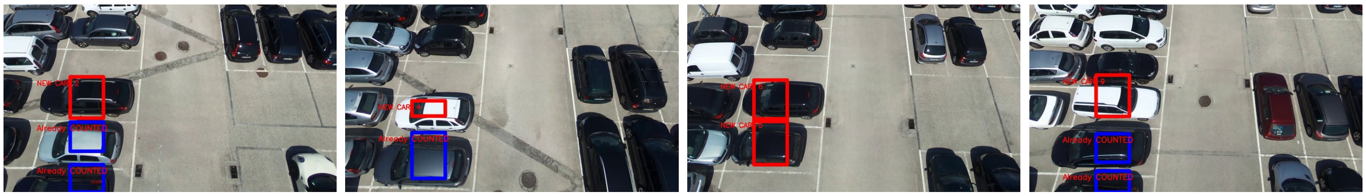

Figure 25. Cars detected on tar parking lot P1.

Figure 26. Cars detected on block pavement parking lot P5.

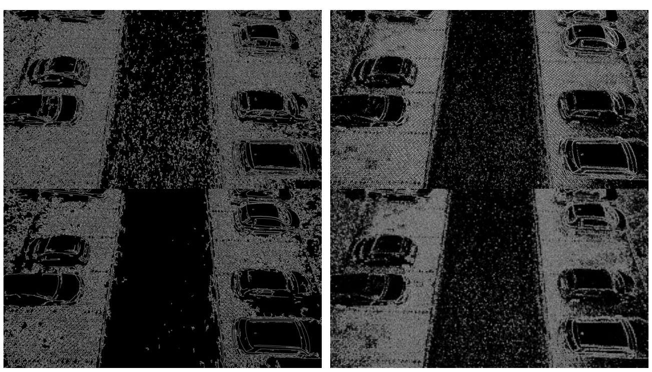

Figure 27. Two different edge detection algorithms tested with different parameters. From left to right, top to bottom: Canny Min=10, Max=50, Sobel X=1, Y=1, Canny Min=2, Max=200, Sobel X=2, Y=2.

TABLE I. RESULTS FOR TAR PARKING LOT.

Parking Lot	Parked Cars	Counted Cars	Repeated Counting	False Positives	Undetected Cars	Detection Rate
P1	17	17	0	0	0	100%

is not easy as there are some homogeneous surfaces in the parking lot borders.

The technique used to detect road borders in the parking lot P5 is based on the grids detection, as referred before. This is very tricky since there is no color differentiation between pavement and grids, all that changes is edges density. Hough Lines detector keeps being used to determine border lines and, as expected, it increases processing time. In Table II detection results are presented while some detection examples are presented in Figure 26.

TABLE II. RESULTS FOR BLOCK PARKING LOT.

Parking Lot	Parked Cars	Counted Cars	Repeated Counting	False Positives	Undetected Cars	Detection Rate
P5	61	61	2	0	2	97%
P5	78	78	1	1	2	97%

In the parking lots 2 and 3, the pavement is mixed. An homogeneous tar surface is found on the road, while the rest is made on block pavement.

Road detection is made based on a sliding window, which runs from the image center to its borders. This window might not move pixel by pixel, it may, for instance, jump 10 pixels every step saving some processing time. On its turn, window size might also be increased, loosing some definition on the road border found but improving the performance of the algorithm.

To evaluate this, several experiments were made in P2 in order to find a good relation between road borders detection accuracy and processing requisites as shown in Figure 24 and Table V.

Since the goal is the development of fully autonomous drones, we tested the developed algorithms on several single-boards (Raspberry Pi 2 Model B, IGEPv2 DM3730 and EPIA-P910) in order to decide what could be the best hardware solution, considering not only processing capacity but also properties like weight and power consumption . Eight random

TABLE III. DETAILED PROCESSING TIME FOR BLOCK PARKING LOT USING A RASPBERRY PI 2 MODEL B.

Frame	1	2	3	4	5	6	7	8
Open Time (RGB)	128	125	131	127	130	127	129	127
Gray	36	22	22	22	22	22	22	22
HSV	135	129	131	133	129	127	134	128
Find road	494	478	498	481	491	483	490	481
Car analysis	226	223	234	227	216	224	218	215
Cars found	1	1	2	2	1	1	1	1
Cars on image	**1**	**1**	**2**	**2**	**1**	**1**	**1**	**1**
%	100%	100%	100%	100%	100%	100%	100%	100%
TOTAL (ms)	1073	1030	1069	1042	1041	1035	1044	1025
				% AVG	**100%**			

TABLE IV. DETAILED PROCESSING TIME FOR MIXED PAVEMENT PARKING LOT USING A RASPBERRY PI 2 MODEL B.

Frame	1	2	3	4	5	6	7	8
Open Time (RGB)	141	144	142	146	140	144	137	145
Gray	36	22	23	23	22	22	22	22
HSV	138	128	130	127	128	130	128	129
Find road	431	430	422	428	421	427	422	420
Canny	284	278	272	281	284	280	277	271
Car analysis	226	225	223	218	222	225	245	237
Cars found	1	1	1	1	1	1	2	2
Cars on image	**1**	**1**	**1**	**2**	**1**	**1**	**2**	**2**
%	100%	100%	100%	50%	100%	100%	100%	100%
TOTAL (ms)	1309	1280	1265	1276	1269	1280	1284	1276
				% AVG	**94%**			

TABLE V. RESULTS FOR MIXED PARKING LOTS.

Parking Lot	Parked Cars	Counted Cars	Repeated Counting	False Positives	Undetected Cars	Detection Rate
P2	46	45	0	1	2	96%
P2	20	20	0	0	0	100%
P3	39	39	0	1	1	97%
P3	24	23	0	0	1	96%

images acquired were chosen and analysed on the singleboard. Processing times for each step of the algorithm detailed before are presented in Tables III and IV. The processing times obviously increase when running the developed algorithms on these single boards. However, we observe that it is possible to reduce the speed of the drone because the images acquired continuously have a considerable repetition of information. With this in mind, and evaluating the experimental results obtained, we consider that Raspberry Pi 2 reaches reasonable values for onboard processing.

VIII. CONCLUSION

The algorithms presented in this paper showed promising results for the detection of vehicles on low altitude images acquired by Drones, being a solution for parking lots management.

The developed algorithms fulfilled the low processing requirements, which enables the algorithms to process images every second and allows the drone to move at a reasonable speed; the accuracy associated to vehicle detection and counting is also high. Furthermore, results obtained for tests made in three different types of pavement indicated a versatile solution, adaptable to several contexts achieving good performances with slight parameter adjustments from park to park.

As future work, we are developing algorithms for boats detection on water and the preliminaries results were also satisfactory. We think this work can provide an interesting contribution to our future smart cities, as a starting point for monitoring of objects of interest using drones. Moreover, we are optimising the presented algorithms to be used onboard of the drone in order to have a fully autonomous solution.

REFERENCES

[1] A. J. R. Neves, M. Camarneiro and L. Cozinheiro. "Vehicle Detection on Low Altitude Images Based on Edge Density," Proc. of the Second International Conference on Advances in Signal, Image and Video Processing, SIGNAL 2017, pp. 3337.

[2] `www.parrot.com/us/Drones/Parrot-Bebop-2`, Bebop 2 drone, retrieved April, 2017.

[3] `www.amazon.com/b?node=8037720011`, Amazon prime air, retrieved April, 2017.

[4] J.-Y Choi and Y.-K. Yang, "Vehicle detection in aerial images," Technical report, College of IT, Kyungwon University, 2009.

[5] `dailytech.com`, Flying rescue drone prototype to provide lifesaving aid to swimmers, retrieved April, 2017.

[6] R. Maini and H. Aggrwal, "Study and comparison of various image edge detection techniques," Technical report, Punjabi University, India, 2009.

[7] S. Tuermer, J. Leitloff, P. Reinartz and U. Stilla, "Automatic vehicle detection in aerial image sequences of urban areas using 3D HOG features," Technical report, Technische Universitaet Muenchen, German Aerospace Center, 2010.