Agronomy Research **14**(3), 733–744, 2016

Technical and software solutions for autonomous unmanned aerial vehicle (UAV) navigation in case of unavailable GPS signal

M. Dlouhy[1], J. Lev[2] and M. Kroulik[1,*]

[1]Czech University of Life Sciences Prague, Faculty of Engineering, Department of Agricultural Machines, Kamýcká 129, CZ165 21 Prague 6, Czech Republic
[2]Czech University of Life Sciences Prague, Faculty of Engineering, Department of Physics, Kamýcká 129, CZ165 21 Prague 6, Czech Republic
[*]Correspondence: kroulik@tf.czu.cz

Abstract. The article presents autonomous navigation for Unmanned Aerial Vehicles (UAV) without GPS support flying in extremely low altitudes (1.5 m – 2.5 m). Solution via visual navigation as an alternative to missing GPS position was proposed. MSER (Maximally stable extremal regions) was used as a navigation algorithm for detection of navigations objects. While GPS is useful for waypoints specification there are scenarios where GPS has unreliable signal (orchards) or is not available at all (indoor machinery halls or greenhouses). For that reason existing installed camera which is already needed for the task of inspection was used. The navigation algorithm was tested in two scenarios. The first experiment was done with dashed line marked on the floor of the hall. 8-loop testing track was created approximately 10 meters long so it was possible to fly it several times. Then outdoor experiments were performed on the university campus and park roads.
One of the discoveries was that MSER algorithm, proposed for finding correspondences between images, is possible to run in real-time. High reliability of the navigation algorithm was found during the indoor testing. The incorrect detection of the dashed line was found only in 1% of cases and those failures did not cause failure of navigation.
Although outdoor road recognition is difficult in general due to various surfaces and smoothness, MSER was able to find suitable candidates. When the UAV was fed with the parameter of road width it could verify that information with estimated distance and camera pose to accept or reject the detected pattern. The road was successfully recognized in 40% cases. Similar to the indoor algorithm in the case of navigation failure navigation along the absolute trajectory (line) was used.

Key words: robot, machine vision, maximally stable extremal regions, algorithm.

INTRODUCTION

Agricultural robots can potentially take the place of manual labour, particularly in performing hazardous tasks such as protection of plants from pests, but also to improve productivity and profit-ability in farms, occupational safety and environmental sustainability. A UAV is an appropriate tool to perform multi-temporal studies for crop monitoring at low altitudes. (Torres-Sánchez et al., 2013). The application of UAVs has many advantages such as ease, rapidity, and cost of flexibility of deployment that makes

UAVs available in many land surface measurement and monitoring applications. For the UAV navigation the ground station and the predefined waypoints are usually used (Xiang, 2011; Gomez-Candon et al., 2014). On the other hand GPS technology has several critical drawbacks including insufficient accuracy for precision agriculture, interruptions in the signal and alterations of the environment which are not in the map but which need to be taken into account. This may lead to navigation failure (Santosh et al., 2014). According to Li et al. (2009) GPS and machine vision fused together or one of them fused with another auxiliary technology is becoming the trend development for agricultural vehicle guidance systems. The navigation of UAVs is still one of the most important subjects in defence research, especially in GPS-denied environments (Michaelsen & Meidow, 2014).

Autonomous navigation of field robots in an agricultural environment is a difficult task due to the inherent uncertainty of the environment (Hiremath et al., 2014). Michaelsen & Meidow (2014) summarized the related work about methods of automatic control of UAVs. Michaelsen & Meidow (2014) reported on the statistical embedding of a structural pattern recognition system into the autonomous navigation of an UAV during simulation of flight. A rule-based system is used for the recognition of visual landmarks such as bridges in aerial views. Fei et al. (2013) presented a comprehensive control, navigation, localization and mapping solution for an indoor quadrotor unmanned aerial vehicle (UAV) system. Three main sensors was used onboard the quadrotor platform, namely an inertial measurement unit, a downward-looking camera and a scanning laser range finder. The UAV, after being issued with the main navigation command, does not need to maintain any wireless link to the ground control station. Babel (2014) considered UAVs equipped with landmark-based visual navigation, a system which is less vulnerable to hostile acts than GPS or to long-term GPS outages, since it is not guided by external signals. A navigation update was obtained by matching onboard images of selected landmarks with internally stored geo-referenced images.

Methods often combine signals from multiple sensors. On the other hand, UAVs carrying capacity is limited. For this reason it is preferred to use existing equipment of the UAV without additional resource load. In the case of navigation through image analysis in real time a large amount of data is processed. According to Matas et al. (2004) the inmost images there are regions that can be detected with high repeatability since they possess some distinguishing, invariant and stable property. These regions may serve as the elements for stereo matching or object recognition. Authors presented the maximally stable extremal regions (MSER) algorithm for an efficient and practically fast detection of objects.

Main aim of this paper is presenting results of MSER algorithm utilization for autonomous UAV flight and navigation. This algorithm has not been used for the UAV navigation yet.

MATERIALS AND METHODS

Robot

The robot used for control algorithms was commercially available quadcopter AR.Drone 2 (AR.Drone 2.0, 2015). It is relatively cheap (approx. 300 EUR) and worldwide available unmanned aerial vehicle. This means that it is easily replaceable in

case of damage and also that other researches can simply repeat the experiments. The UAV is meant for mass market and that requires high safety for human-robot interaction.

The robot is controlled via Wi-Fi. You can operate it with FreeFlight2 application for tablets or mobile phones, but there exists also SDK (Software Development Kit) available for developers. The API (Application Programming Interface) is open to public and provides opportunities even for scientific research.

There are two cameras on the robot. The front camera has resolution 720p at 30 fps, wide angle lens 92°. The down pointing camera has lower resolution, but runs at 60 fps and it is used primarily for vehicle stabilization. They cannot be used simultaneously, but the operator/program can switch from one video stream to another anytime during the flight. The AR.Drone 2 is powered by LiPo 1,000 mAh battery (available is also bigger 1,500 mAh) which is enough for approximately 10 minutes long test flights.

The robot is equipped with three axis accelerometer, gyros and compass. Moreover it has down pointing sonar for low altitude and barometer for higher altitude control. The robot stabilization is fully handled by on-board computer and user sends only macro commands like ‘take off’, ‘set desired pitch, roll and yaw’ or ‘land’. The robot responds with sensors status messages. It is possible to switch to DEBUG mode when information from all sensors and internal estimates including absolute 3D position and 3D angles are transmitted.

Indoor experiment

There was a track created in the form of figure eight on area 5 × 10 m. It was in the indoor hall for the purpose of development and testing of the first algorithm. For the navigation dashed black line on light floor background was used. There were placed two wooden columns in the centre of the Fig. 8. The whole setup is sketched on Fig. 1. Note, that this test track is similar to Air Race competition, which has been part of Robot Challenge/Vienna since 2012 (RobotChallenge, 2016).

The prerequisite for presented project was implementation of framework handling communication with the ARDrone2 robot. The code was written in Python and the control program can run on any device with Wi-Fi connection. The frameworks as well as presented algorithms are freely available on GitHub (Dlouhý, 2015).

Navigation Algorithm

The autonomous navigation code requires two processing threads. The first thread is main control loop running at 200 Hz handling communication and commands for the robot in real time. The second (working) thread is processing video stream and feeds image result data into the first control thread. The control loop simultaneously handles P-controller (proportional controller) for desired height and forward speed. The trajectory is defined by line or circular segment of given radius. Again simple P-controllers are used for angle and offset correction to navigate along the curve/line. The placement of navigation segment is updated with every processed video image.

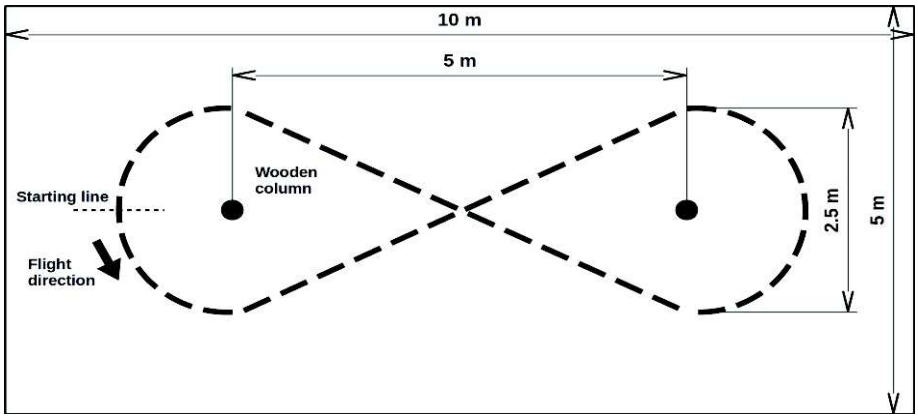

Figure 1. The testing track created in form of figure eight.

The image processing thread receives video stream and converts I-frames (Intra-coded picture, coded without reference to any frame except themselves, (Wiegand et al., 2003) into RGB picture. The supporting library for image processing is OpenCV 2.4.8 (package cv2) with binding to Numpy 1.8.2 (package for scientific computing with Python). The set of necessary cv2 functions was relatively small. First of all, cv2.MSER class for 'Maximally stable extremal region' extractor was created. This method was first published by Matas et al. (2002). MSER is a way how to overcome cumbersome definition of threshold for grayscale image segmentation. The basic idea is to explore all possible thresholds at once. If you imagine animation of threshold from 0 to 256 you will get white image at the beginning, then darkest features will appear and you end up with black image. On every step you could do analysis of black or white 'objects' and measure their area. The output of MSER are maximally stable objects, i.e. objects which do not change much over slight change of threshold.

There are several parameters for MSER method primarily to reduce the number of detected objects. Required is δ parameter (threshold step) and minimum and maximum size (area) of detected objects. Successful results were obtained for parameters $\delta = 10$, minimum area = 100 pixels and maximum area = 30,000 pixels.

When objects are segmented function cv2.minAreaRect is used to find a rotated rectangle of the minimum area enclosing the input 2D point set. Rectangle was only accepted if points covered more than 70% of the area. These basic functions were then followed by several filter procedures. First of all duplicities were removed, i.e. overlapping rectangles for different threshold values. The one which was closer to desired width/height ratio was used and the other was rejected. The second filter removed the square like rectangles (when the length was less than three times the width), and width itself had to be in given limits (75 to 200 pixels, corresponding to expected height above the ground).

The result of image processing was list of filtered rectangles with their coordinates within the image, width and height, and angle orientation. These data were integrated into navigation control program with projection to 2D, where the altitude of the robot was estimated by maximal width of remaining rectangles. This pose scaling was much more robust than evaluating height for each rectangle separately.

The ARDrone2 has two video channels: one for quick overview with possibility of lost frames and one for video recording with extra several MB large buffers on board. It was decided to use recorded stream because otherwise it could miss some important crossing during temporary connection failure. The price for this was that the video channel was a little bit delayed (typically up to 1 second, where longer times were reported to operator as potential danger). Note, that only I-frames were used from H.264 video codec.

The delayed video stream with extra delay caused by image processing was handled with 'pose history queue'. The basic update runs on 200 Hz so it was enough to remember 200 poses for the fast history lookup. This way absolute coordinates of navigation strips were available.

The set of detected strips defined segment on which the UAV should navigate within next second (until a new image was received). A simple state machine was used to distinguish between navigation along the line, clockwise and anticlockwise circle. If there was only one segment/strip/rectangle in the processed image, then the previous state was kept. Two and more strips then defined line or circle of fixed radius.

There is one situation which requires a special care: trajectory path self-crossing. The navigation algorithm has to select subset of relevant strips/marks. In particular due to the narrow FOV (Field of View) of the down pointing camera it could see only two strips of crossing line.

First of all pairs are analysed if they define consequent line or circle. The first strip defined coordinate system and the second had to be x in range 0.25 m to 0.48 m, y (absolute offset to the left/right) in -0.25 m to +0.25 m and finally angle change had to be less than 50°. If there exists such a pair it was verified against current robot position. For classification as line and angle difference bigger than expected crossing angle was pair rejected.

Second, if no pair was found, only individual strips were compared against the strip poses from the last processed image. If again no suitable pair was found, then search for identical strips from the current and the previous image was performed for update path absolute position only. As the last step list of the detected strips was updated reference line or reference oriented circle was defined. One more note about speed control. The navigation pattern (number of strips on the floor defining line and arcs) was known so after state transition the control algorithm could increase the speed for given number of segments and slow down as expected line-arc or arc-line transition was approached. Moreover in case of communication problems when video was delayed for more than 2 seconds the speed was reduced to zero.

Outdoor experiment

The goal was to present navigation on visually distinctive features like paved road. The algorithm for outdoor navigation along natural landmarks was slightly different but used the same core. The front camera was used instead of the down pointing camera for better situation overview. Because the road as the primary 'navigation line' is on the ground, only the lower parts of images were used. The road from perspective view is no longer rectangle. Individual horizontal image strips were processed via MSER where convex hull was applied on detected objects. A simple approximation via trapezoids was used.

The image trapezoid corners were projected on the ground, based on the history poses as for indoor experiments and the width of potential detected road was computed. For absolute camera position it was necessary to remember all 3 Euler angles (pitch, roll and yaw) and also estimate of UAV altitude (distance from the ground).

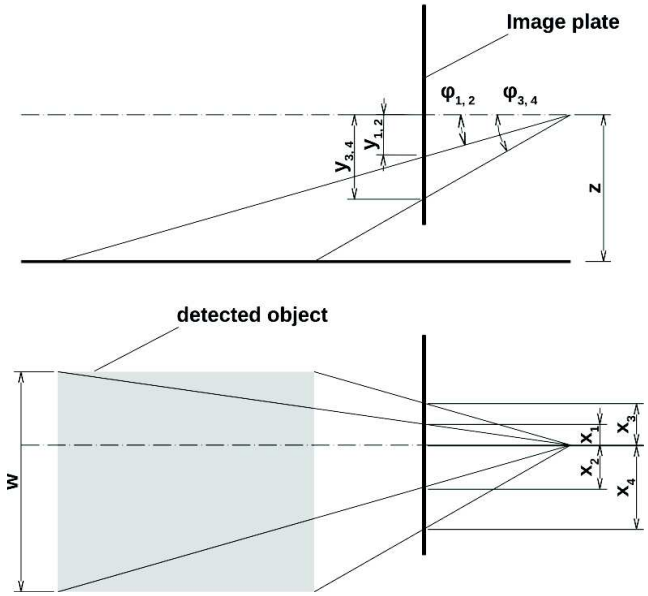

Figure 2. Side view and top view of simplified reverse projection from image plane to the ground. w – width of the detected road, x_i y_i – image coordinates, φ_i – angles between axis of robot and connecting line between robot and point on the detected object.

Fig. 2 shows simplified reverse projection of image plane points to the ground. The input parameters are estimated UAV absolute coordinates (6D, including angles) and simplified camera model defined by image centre (x_c, y_c), FOV (Field of view), and image resolution. There are four image coordinates (x_i, y_i), where I = {1, 2, 3, 4} defines boundary of trapezoid strip. Every image point is converted by following formulas. First image coordinates (x_i, y_i), are rotated along the image centre:

$$x_h = (x_i - x_c) \cdot cos(\alpha) - (y_c - y_i) \cdot sin(\alpha) \quad (1)$$

$$y_h = (x_i - x_c) \cdot sin(\alpha) + (y_c - y_i) \cdot cos(\alpha) \quad (2)$$

where: x_h, y_h – compensated image coordinates; α – roll angle.

In the next step the angles for horizontally aligned coordinates were computed:

$$\psi = k \cdot x_h + \gamma \quad (3)$$

$$\varphi = k \cdot y_h + \beta \quad (4)$$

where: k – ratio of FOV and resolution; β – is pitch angle; γ – robot heading.

Equations (3) and (4) are only simplification because proper equation should take non-circular surface and distortion into account. The ground distance from current UAV position (x, y, z) is then:

$$d = \frac{z}{tan(-\varphi)} \tag{5}$$

where: d – ground distance from current robot position; z – distance from the ground (UAV altitude).

The absolute coordinates of the robot can be calculated:

$$x_b = x + d \cdot cos(\psi) \tag{6}$$

$$y_b = y + d \cdot sin(\psi) \tag{7}$$

where: (x_b, y_b) – path boundary point.

Because the four ground points do not necessarily form any regular trapezoid four different estimates were calculated (each corner against the line on opposite side). The post-processing filter of detected potential road segments was much simpler when compared to indoor experiment where exact size and dash line pattern was known. Here only expected road width was used to select the best matching object. Moreover minimum/maximum and variation limits were used for acceptance. The global navigation algorithm accepted new trajectory line if it fit within given angle (20°) and offset (2 m). The speed control for outdoor line navigation was simplified to slow down only in cases large video delays. Fig. 3 presents image processing loop for indoor and outdoor experiment.

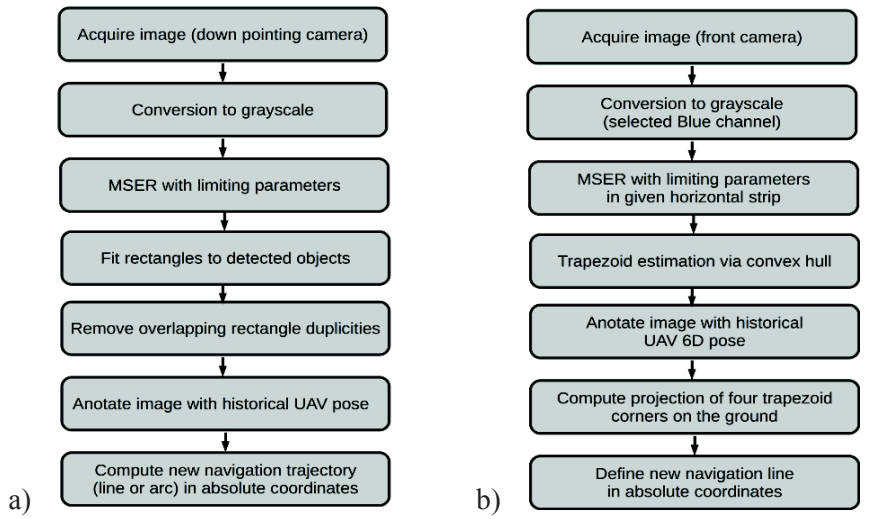

Figure 3. Diagram of the entire image processing: a) indoor experiment; b) outdoor experiment.

RESULTS

Indoor

The final results of the indoor experiment were quite satisfactory. The robot was capable to follow marked route and the limiting factor become battery power. We can state high robustness of MSER algorithm. In particular it is suitable solution for light changing conditions. See some representative examples of problematic cases presented on Fig. 4.

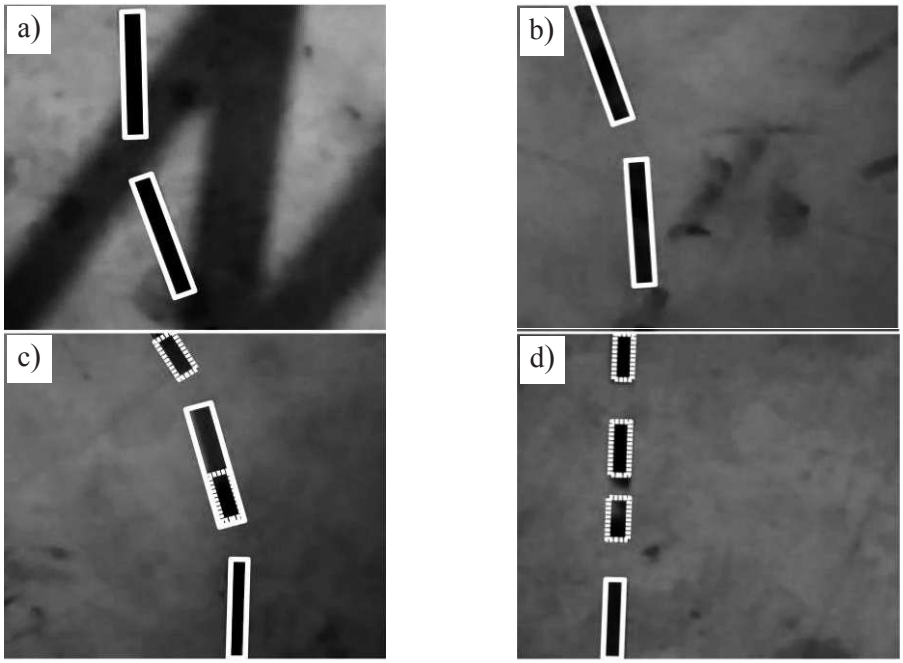

Figure 4. a) – example of strong sunlight and shadows; b) – dirt on the floor; c) – non-uniform colour of some marks due to light angle reflection; d) – broken mark due to strong light reflection.

The success rate was based on evaluation of three testing flights. The total length of selected flights was 20 minutes and 41 seconds. The route marked on the floor was circled in total 27 times. Fig. 5 shows histogram how many strips were typically detected in each video frame. It is apparent that mostly two strips were detected. The presented algorithm requires at least two detected strips for successful operation. This means that in 84.2% the algorithm had ideal relevant data (two or more detected mark in one image) and in 5.9% case it had to use history (no detected mark). The single mark cases (10%) can handle correction of absolute position of the navigation line or circle, but they fail to detect line-circle and circle-line transition.

Note that the robot was capable to navigate along the last defined navigation curve even in spite of detection failures in several consequent frames. This was possible thanks to reliable position estimation (for short term) using UAV inertial navigation unit integrated with optical flow (handled by on-board firmware).

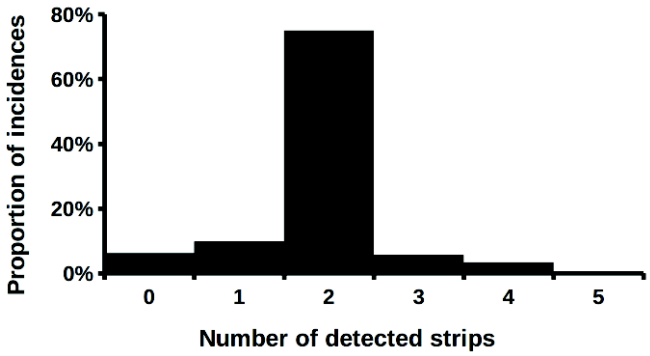

Figure 5. Histogram, number of strips which were typically detected in each video frame.

Important data are also incorrect detections, i.e. detection of false marks. There were only 27 wrong detections (i.e. approximately one per loop, mostly caused by dark pole holder). Correct detection was in 2,420 images i.e. there was approximately 1% failure rate. Further analysis showed that false mark detection were mostly (23 of 27) in combination with two correct marks and 3 times with one correct mark. In one case there was detected only one false mark without correct mark. All cases were successfully rejected by crossing detection procedure.

Outdoor

The first outdoor experiments were performed on the university campus and park roads. Although road recognition is difficult in general due to various surfaces and smoothness, MSER was able to find suitable candidates. When the UAV was fed with the parameter of road width it could verify that information with estimated distance and camera pose to accept or reject the detected pattern.

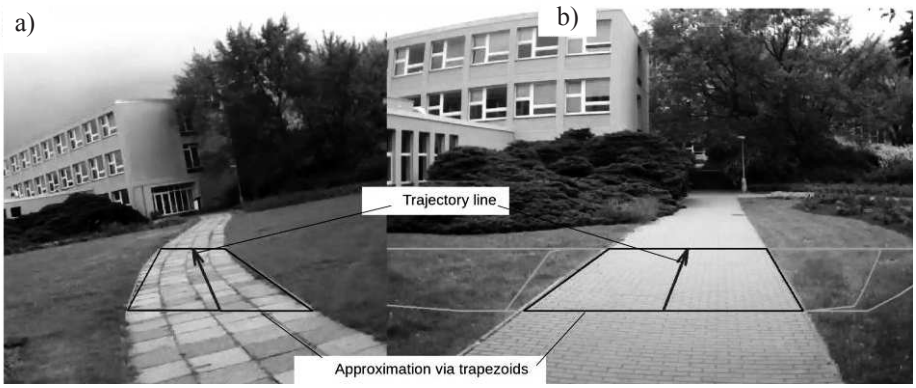

Figure. 6. A sample view of a) – curved and b) – straight road.

There are only relatively short road segments between junctions. The test runs were performed near the Faculty of Engineering on 42 m and 51 m long segments. The width of roads was 3 m and 2.15 m respectively. The first line was solid straight while the second has slight S-shape. The flights were short, 54 s and 106 s including take-off and

landing, and the number of processed images corresponds to number of seconds. Fig. 6 shows sample view of curved and straight road.

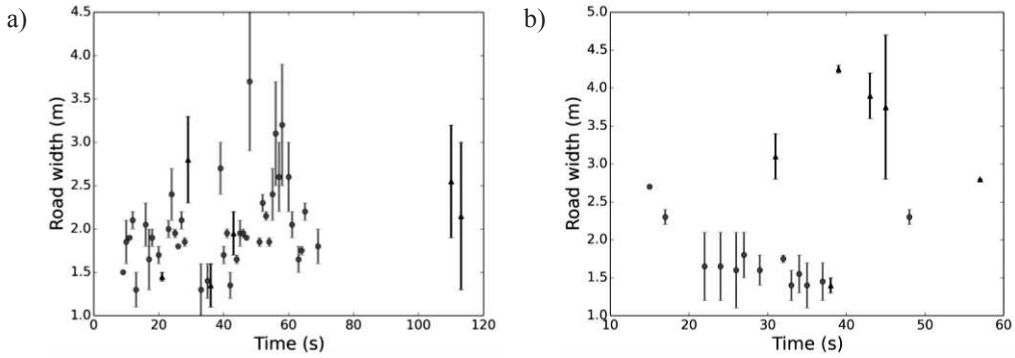

Figure 7. Accepted road widths with 4-corner distances variation. Correctly detected – circle, wrongly detected – triangle. (a) – curved path and (b) – straight road.

Fig. 7 shows correctly (circle) and wrongly (triangle) detected road with its minimum/maximum variation caused by distance measurement from four projected corners. In the last third the road was not visible, so failures are expected. Note that in b-case with straight road lower limit for maximal accepted road width could significantly reduce false detection.

DISCUSSION

Control adaptation and path tracking are essential issues for moving along a crop in an autonomous way, due to the stochastic conditions inherent to crop environments (Urrea & Muñoz, 2013) and the agricultural robots must be able to adapt themselves in response to various terrain conditions (Mahadhir et al., 2014). Path detection during outdoor test was more challenging. Also Michaelsen & Meidow (2014) confirm that flying a UAV with an experimental system is expensive, risky, and legally questionable. These problems can be solved by defined marks. This was the purpose of addressing in the first phase of testing. Although the predefined marks were used for navigation, conditions were not easy for detection. It was necessary to solve the problems that represent light reflections, shadows and dirt on the floor (see Fig. 4). Advantages of the MSER algorithm were demonstrated in this case.

During the indoor experiments high algorithm reliability for navigating UAV was demonstrated. Wrong detection of individual marks ranged around 1%. However, these erroneous detection did not cause an error in navigation. Good applicability relatively inexpensive and commercially available platforms ARdrone2 was also demonstrated.

The outdoor road was successfully recognized in 40% of cases. Similarly to the indoor algorithm in case of navigation failure navigation along the absolute trajectory (line) was used, and successful detection only updated absolute coordinates of the road. The test was performed with single image strip only. The main source of failures was wind, for which the UAV had to compensate, and also angle changes of speed control. This has much bigger effect visible in front camera when compared to marks detection

of down pointing camera. Calculating the width of the road was essential for assessing the accuracy of detection. However, the accuracy of this calculation was significantly influenced by the precision determination of pitch angle (tilt of UAV). Two degrees caused width estimation change by 16% (2.68 m instead of 3.1 m). This error progressively increases depending on the pitch angle error. This error can be easily caused by slight push on UAV foam protective hull during battery exchange and would require extra self-calibration. Less restrictive post-filtering can be an alternative.

CONCLUSIONS

This article presented experiments with UAV flying in extremely low altitudes (1.5–2.5 m). Test included outdoor and indoor environment. The first experiment was performed with artificial landmarks – dashed line marked on the floor of machinery hall. Highly reliable algorithm based on MSER image processing was demonstrated. Algorithm did not fail during the entire course of the tests and the battery power becomes the limiting factor.

In the second part the outdoor application, where the UAV were capable of following visually distinctive patterns, was demonstrated. The MSER image pre-processing was applied to horizontal image strip to recognize road on university campus. In the outdoor environment, it was necessary to distinguish between navigation failure due to an erroneous detection or weather conditions. Weather conditions play a greater role in navigation.

ACKNOWLEDGEMENTS. Supported by the GA TF, Project No.: 2015:31160/1312/3111, Autonomous measuring platform for work in field conditions.

REFERENCES

AR.Drone.2.0. 2015. [online]. *http://ardrone2.parrot.com.*

Babel, L. 2014. Flight path planning for unmanned aerial vehicles with landmark-based visual navigation. *Robotics and Autonomous Systems* **62**(2), 142–150.

Dlouhý, M. 2015. Heidi. GitHub repository. *https://github.com/robotika/heidi.*

Fei, W.A.N.G., Jin-Qiang, C.U.I., Ben-Mei, C.H.E.N. & Tong, H.L. 2013. A comprehensive UAV indoor navigation system based on vision optical flow and laser FastSLAM. *Acta Automatica Sinica* **39**(11), 1889–1899.

Gomez-Candon, D., Labbé, S., Virlet, N., Jolivot, A. & Regnard, J.L. 2014, High resolution thermal and multispectral UAV imagery for precision assessment of apple tree response to water stress. In: *2. International Conference on Robotics and associated High-technologies and Equipment for Agriculture and Forestry RHEA*. Madrid, Spain, pp. 283–288.

Hiremath, S., van Evert, F.K., ter Braak, C.J.F., Stein, A. & van der Heijden, G. 2014. Image-based particle filtering for navigation in a semi-structured agricultural environment. *Biosystems Engineering* **121**, 85–95.

Li, M., Imou, K., Wakabayashi, K. & Yokoyama, S. 2009. Review of research on agricultural vehicle autonomous guidance. *International Journal of Agricultural and Biological Engineering* **2**(3), 1–16.

Mahadhir, K. A., Tan, S. Ch., Low, Ch. Y., Dumitrescu, R., Amin, A.T.M. & Jaffar, A. 2014. Terrain classification for track-driven agricultural robots. *Procedia Technology* **15**, 776–783.

Matas, J., Chum, O., Urban, M. & Pajdla, T. 2002. Robust wide baseline stereo from maximally stable extremal regions. In: *Proc. of British Machine Vision Conference*. Cardiff, UK, 384–396.

Matas, J., Chum, O., Urban, M. and Pajdla, T., 2004. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing* **22**(10), 761–767.

Michaelsen, E. & Meidow, J. 2014. Stochastic reasoning for structural pattern recognition: An example from image-based UAV navigation. *Pattern Recognition* **47**(8), 2732–2744.

RobotChallenge. 2016. [online]. *http://www.robotchallenge.org*.

Santosh, A., van der Heijden, G.W.A.M., van Evert, F.K., Stein, A. & ter Braak, C.J.F. 2014. Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter. *Computers and Electronics in Agriculture* **100**, 41–50.

Torres-Sánchez, J., López-Granados, F., De Castro, A.I. & Peña-Barragán, J.M. 2013. Configuration and specifications of an Unmanned Aerial Vehicle (UAV) for early site specific weed management. *PLoS ONE* **8**(3), e58210.

Urrea, C. & Muñoz, J. 2013. Path Tracking of Mobile Robot in Crops. *Journal of Intelligent & Robotic Systems* **80**(2), 193–205.

Wiegand, T., Sullivan, G.J., Bjøntegaard, G. & Luthra, A. 2003. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology* **13**(7), 560–576.

Xiang, H. & Tian, L. 2011. Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV). *Biosystems engineering* **108**(2), 174–190.