IEEE Copyright Notice

Low-Cost Embedded System for Relative Localization in Robotic Swarms

Jan Faigl, *Member, IEEE,* Tomáš Krajník, Jan Chudoba, Martin Saska

***Abstract*— In this paper, we present a small, light-weight, low-cost, fast and reliable system designed to satisfy requirements of relative localization within a swarm of micro aerial vehicles. The core of the proposed solution is based on off-the-shelf components consisting of the Caspa camera module and Gumstix Overo board accompanied by a developed efficient image processing method for detecting black and white circular patterns. Although the idea of the roundel recognition is simple, the developed system exhibits reliable and fast estimation of the relative position of the pattern up to 30 fps using the full resolution of the Caspa camera. Thus, the system is suited to meet requirements for a vision based stabilization of the robotic swarm. The intent of this paper is to present the developed system as an enabling technology for various robotic tasks.**

I. Introduction

This paper presents a simple to use, small, and light-weight vision based embedded system for a relative localization within a robotic swarm. In particular, the developed system is motivated by practical needs to stabilize a swarm of *micro aerial vehicles* (MAVs) and it is aimed to fulfill requirements of robotic swarms acting in real world outdoor environments.

The system is based on a detection of black and white (B/W) pattern with a precision in units of centimeters for distances in units of meters. The system provides estimation of the relative position up to 60 Hz and may be directly employed in a feedback loop of swarm control and its shape stabilization. Moreover, it is worth to mention the system is made of low-cost off-the-shelf components, which facilitate its reproducing as is required by the ideas of swarm research.

The main intention of this paper is to present the system, its model, and real performance characteristics with respect to the intended application. Algorithms, that ensure the swarm shape and coordinate the MAVs, must take into account operational constraints, describing where neighbouring swarm particles (equipped with the pattern) provide localization with expected precision and reliability. Considering a model of localization precision may significantly decrease the overall position estimation uncertainty and increase reliability of the autonomous system as it is shown in our previous work on this topic [1]. The constraints also support users decisions on a proper camera settings based on the expected application. Therefore, we present a model of the localization arising from theoretical analyses of the vision system and experimental evaluation of the system performance in real scenarios with regard to its practical deployment.

Authors are with Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27 Prague, Czech Republic {faiglj,krajnt1,chudobj,saskam1}@fel.cvut.cz,

II. Related Work

The relative localization presented in this paper is based on detection of a geometric pattern that is carried by an object (e.g., a neighbouring MAV in the swarm) to which the robot carrying the camera module has to be relatively localized. The problem of geometric pattern detection is one of the fundamental issues studied in computer vision. A basic approach to address this problem is the Generalized Hough Transform [2] for finding parameters of the expected geometrical shapes. In particular, we are interested in a circle (ellipse) detector, i.e., the Fast Circle or Ellipse Hough Transform, as its detection can be faster than detection of an arbitrarily shaped object. However, it is known fact that this approach is computationally demanding, and therefore, many approaches have been proposed to address this limitation.

For example authors of [3] consider the RANSAC algorithm on a set of feature points extracted by the Canny edge detector. In [4], the problem is restricted to objects contrasting with the background. Although many of the proposed approaches speed up the finding process and are often called real-time, the considered computational resources are still more powerful (e.g., a regular workstation) than small embedded devices that can be placed on MAV. This is also the case of alternative approaches that try to consider additional geometric properties of the expected circle (ellipse), e.g.,[5], [6], or dividing the problem into sub-problems [7].

Moreover, systems that have been implemented using embedded platforms and providing real-time capability assume conditions that are not satisfied in scenarios we are considering. For example a system running at 12.5 Hz based on detection of pixel colors presented in [8] is not suitable for varying lighting conditions.

On the other hand, the problem of the relative localization using on-board image processing directly at MAV is investigated within the context of safe autonomous landing and take-off. In [9], which is probably the closest approach to our proposed solution, authors consider a set of concentric white rings and their detection using standard algorithm in the OpenCV. Although they report up to 100 Hz image processing frequency for the resolution 640×480 and desktop CPU at 2.4 GHz, the reported frequency is only 10 Hz for a Gumstix Verdex board and the resolution 320×240. The further approaches for stabilizing MAV hovering above the landing pad also consider a relatively simple pattern for the detection [10], [11]; however, they are out of our interest because they utilize a ground station for the image processing and wireless video transmission.

A recent low-cost system for localization of MAVs has been presented in [12]. The authors use the Gumstix Overo Fire computer running at 600 MHz accompanied by the PointGrey FireFly USB2.0 camera. The localization is based on a detection of a pattern with four orange tennis balls from a flying MAV. Even though authors report on-board processing with 60 Hz for the resolution 640×480, the system needs installation of the pattern. Moreover, it has been evaluated only in an indoor environment with stable lighting conditions, and therefore, it does not meet our requirements.

Another approach for MAV with 60 Hz on-board image processing of a landing pad is proposed in [13]. The pad contains a pattern forming a letter 'H' surrounded by a black ring. The detection of the pattern utilizes edge detection algorithm (Canny detector) and ellipse fitting approach [14]. The results presented in the paper are very promising; however, the considered MAV is equipped with Intel Core 2 DUO running at 1.86 GHz, which provides significantly more computational power, especially in floating point operations needed in the used pattern detection mechanism, than the small and power efficient Gumstix Overo. In contrast, our solution provides image processing at 500 Hz for the same resolution and a single core desktop CPU running at 2 GHz.

Based on the available solutions we consider a simple pattern formed from two concentric circles to achieve a good trade-off between detectability, precision of the estimation, and computational requirements. Contrary to the method [15], we rather consider an approach that does not rely on a floating point computational unit, because of our preference of a power efficient ARM processor. The proposed localization system differs from the aforementioned approaches mainly in the following aspects. It is based on light-weight, small, and low power consumption off-the-self components while it provides estimation of the relative position up to 30 Hz for the resolution 640×480. It is clear that the above described approaches, e.g., based on standard image processing techniques with ellipse fitting, can be used for the addressed problem; however, the uniqueness of the presented system is in its direct focus on practical deployment in a robotic swarm operating in an outdoor environment with regard to be easy to use by end-users.

III. System for Relative Localization

The presented system for the relative localization consists of hardware electronic boards accompanied by software for detecting the pattern and a client library for reading the estimated position of the detected blob. The hardware is a standalone camera module accessible through a WiFi connection. The connection to the localization data is realized using a custom protocol and supplementary software library.

The hardware components are based on off-the-shelf components and the software is available on-line. Therefore, the presented localization system can be easily reproduced and deployed in other tasks. All the results and performance characteristics of the localization system presented in this paper have been collected using exactly the hardware components and the pattern detector described bellow.

A. Hardware Components

The core of the module is the Gumstix Overo board with the OMAP 3503 Application processor running at 600 MHz and accompanied with 128 MB RAM and 802.11b/g wireless communication. The second board is the Caspa camera board with the Aptina MT9V032 CMOS sensor. This board is also an off-the-shelf product provided by Gumstix.

TABLE I
Hardware specification

Paremeter	Values
CPU:	OMAP3503 @ 600 MHz running armv7l GNU/Linux 2.6.34
CMOS sensor:	Aptina MT9V032, 752×480 @ 60 Hz
Lens:	F=2.8, FoV$_{min}$ 42°, IR cut filter
Overo Board:	58 mm × 17 mm × 5 mm, weight 6 g (including μSD card)
Camera Board:	39 mm × 26 mm × 25 mm, weight 22.9 g (with lens)
Minicom Board:	60 mm × 19 mm × 7 mm, weight 6 g
Voltage Regulator Effic.:	89 % (cpu idle), 84 % (cpu full load)
Power Consumption:	$\leq$ 2.6 W (2.39 W cpu idle, 2.55 W tracking at full speed)

In addition to these boards, a custom interface board called Overo Minicom is developed to provide similar functionality like e.g., Gumstix Pinto board but having smaller dimensions. A summary of the hardware parameters is shown in Table I.

B. Principle of Pattern Detection

A principle of the blob detection is based on an image segmentation and finding two discs forming a black and white ring placed at a white background, see example of the pattern in Fig. 1.

First, an image acquired from the camera is converted to a binary image using a threshold value θ, which separates bright and dark pixels. Then, continuous segments of dark pixels are searched using a region growing technique. As soon as the region growing is finished and a complete segment is found, its center, bounding box dimensions and area (i.e., the number of pixels) are calculated. If the segment area is approximately (we use a 30 % tolerance margin) equal to the expected value calculated from the bounding box dimensions, the region growing algorithm jumps to the segment center and starts searching for a continuous segment of bright pixels. Whenever a bright segment is found, its area, bounding box dimensions, and center are calculated as well. If the bright segment area corresponds to its bounding box dimensions, the dark and bright segments are tested for concentricity and the ratio of their areas is calculated and compared with an expected value. In the case the segments pass the tests, it is assumed that they represent the searched pattern, otherwise it is reported that a valid blob has not been found.

Once the pattern is detected, the values of the mean brightness m_d and m_b of the pixels in the dark and bright discs, respectively, are computed and the threshold θ is set to $(m_d + m_b)/2$. If the pattern is not detected, the threshold θ is either increased or decreased.

C. Principle of Pattern Localization

The data gathered in the detection step are used to determine the pattern relative position to the camera module. We assume the radial distortion is not extreme and the pattern appears in the captured image as an ellipse. To determine the pattern distance, we exploit the fact, that the length of the principal axis of this ellipse is invariant to the pattern spatial orientation and depends solely on its position relative to the camera. To obtain principal axis length and orientation, we calculate a covariance matrix of pattern's pixel positions and compute its dominant eigenvalue and eigenvector. Then, we calculate spatial coordinates of the principal axis endpoints $\lambda\mathbf{s_a}$ and $\lambda\mathbf{s_b}$ up to the unknown factor λ. Since $|\lambda\mathbf{s_a}, \lambda\mathbf{s_b}| = d_o$ we can obtain $\lambda = d_o/|\lambda\mathbf{s_a}, \lambda\mathbf{s_b}|$. The pattern position $\mathbf{x}$ is then calculated by as $\mathbf{x} = \lambda\left(\mathbf{s_a} + \mathbf{s_b}\right)/2$. Note, that the aforementioned calculation determines principal axis length with a subpixel precision.

D. Computational Complexity and Implementation Notes

The computational complexity of the above described detection and relative localization of the pattern mainly depends on the image resolution. The most computationally intensive part is the image thresholding, which needs two processor cycles per pixel. The finding procedure can be repeated until the whole image is processed, and several blobs can be found. However, in a case of the single blob, the searching process can be terminated once the first blob is detected. Moreover, the searching procedure can be speeded up using the previous position (center) of the blob, which represents a kind of *tracking* mechanism. On the other hand, if the pattern's position is changed significantly between two consecutive images, the whole image has to be processed.

Although, we can express the computational complexity as a proportional $O(n)$ to the number of image pixels n, the real computational requirements are affected by the constant factor, which depends on the number of the CPU cycles per pixel. Since the region growing operation takes at least four additional operations per pixel, the tracking mechanism can improve the processing time triply. The real performance gain of the tracking is presented in Section V.

IV. Sensor Model

In this section, we present a model that provides basic information about the expected performance of the sensor. This can be helpful for end-users to choose resolution and diameter of the pattern according to particular application scenario and expected requirements, e.g., required fps, coverage, and precision of the localization. These two parameters can be principally selected, as the other parameters depend on the properties of the system components, which may not be adjustable for the end-users.

Regarding the practical deployment of the relative localization system in stabilization of the robotic swarm, the most critical property of the localization system is its "operational space" or its coverage, i.e., a space where the pattern is reliably detected and localized. The dimensions of the operational space are given by the camera focal length, image resolution, pattern spatial orientation and diameter.

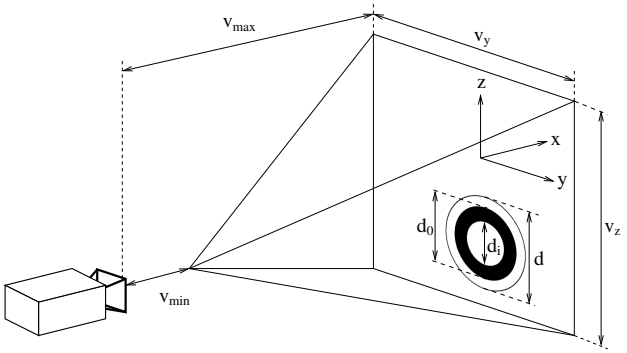

Fig. 1. Geometry of the pattern and the operational space.

Considering an ideal pinhole camera model, the operational space has a pyramidal shape with its apex closer to the camera, see Fig. 1. The image dimensions w_i, h_i of the pattern can be calculated by

$$w_i = f_x d_o \cos(\varphi)/x, \qquad h_i = f_y d_o \cos(\psi)/x,$$

where x is the pattern distance from the image plane, d_o is the pattern diameter, φ and ψ represent the pattern pitch and roll and f_x, f_y are the camera focal lengths.

Since the pattern has to be completely within the camera field of view and its pixel width and height must exceed a value D (experimentally found as D=11), the maximal v_{max} and minimal v_{min} detectable distances are

$$\begin{aligned} v_{min} &= d_o \max\left(f_x/w, f_y/h\right), \\ v_{max} &= d_o/D \min(f_x\cos(\varphi), f_y\cos(\psi)). \end{aligned} \tag{1}$$

where w and h is the image horizontal and vertical resolution in pixels, respectively.

Having v_{max}, the dimensions of the base of the operational space v_y and v_z can be calculated as

$$v_y = v_{max} f_x/w - 2d_0, \qquad v_z = v_{max} f_y/h - 2d_0. \tag{2}$$

Using the aforementioned equations, a sufficient diameter of the detected pattern d_o can be established out of the required operational space size and pattern orientation restrictions.

V. System Performance Characteristics

The performance of the developed localization system has been experimentally verified in a series of real scenarios in which the system meets requirements arising from its practical deployment for a relative localization in a robotic swarm. The performance metrics are the operational frequency of the provided estimation of the relative position, the maximal distance of the detectable pattern, an expected error of the distance estimation, and the operational space restricting the camera field of view for a reliable blob detection.

All herein presented experimental results have been achieved using the hardware described in Section III-A; thus, they represent real achievable results in a practical deployment of the developed localization system. Results presented

in Section V-A and Section V-B have been measured using a pattern with d_o=14 cm, d_i=8.4 cm, and d=18 cm.

Evaluation Methodology – The real performance of the relative localization system has been verified using the developed module providing information about the tracked object by the UDP server and WiFi connection. All the parameters of the sensor model have been identified and adjusted; hence, the presented results represent real achievable performance metrics for end-users. The estimated position can vary due to small changes in the captured images, and therefore, average values of the valid estimations (if the blob has been detected) over a period (typically for 100 measurements) are considered. In fact, the pattern has been successfully detected for several such periods without any fall outs.

A. Real Computational Requirements

The real computational requirements depend on the size of the blob and successful detection of the blob using its previous position (a.k.a. tracking), see Section III-D. Therefore, the requirements have been experimental identified using the developed system. They are measured as the maximal number of the processed images per second (herein denoted as FPS following standard conventions) in the full processing loop, i.e., including the capturing time, YUV to grayscale conversion, blob detection, transformation of the coordinates using camera parameters, and transfer of the coordinates from the camera module to a client computer using UDP protocol over WiFi. Hence, the presented frequencies demonstrate a real application and really achievable FPS.

The blob detection has been implemented in C++ and compiled by the G++ ver. 4.3.3 cross-compiler for the `arm-angstrom-linux` distribution used on the Gumstix Overo board. The detection considers only B/W pattern, therefore a grayscale image from the Caspa device is used to reduce the computational burden.

Although the image conversion and coordinates transformation depend only on the image resolution, the blob detection itself can vary. First, the tracking of the blob can significantly reduce the required time to find the expected segments. In addition, the processing time also depends on the number of pixels forming the segments, i.e., a smaller pattern or a pattern at a longer distance can be detected faster than a larger pattern or a pattern close to the camera. The worst case scenario is a situation when a blob is not detected, because it requires searching of the whole image.

TABLE II
REAL COMPUTATIONAL REQUIREMENTS - FPS

Resolution	L_{max} [m]	FPS_{worse}	FPS_{max}
320×240	3.2	33	60
480×360	3.5	18	46
640×480	5.5	9	30
752×480	5.5	7	27

Regarding these aspects the achievable FPS is measured for four selected resolutions (see Table II) and for a pattern placed at the camera optical axis at the distance L from the camera. The guaranteed FPS is depicted in Table II, where FPS_{worse} denotes the situations when the blob is not detected at all and the whole image must be searched. The L_{max} column denotes the maximal distance at which the blob is reliably detected, i.e., the blob is continuously detected without failure for a couple of seconds. The column FPS_{max} denotes maximal image processing frequency when the blob with $L > 1$ m is perfectly found using the tracking.

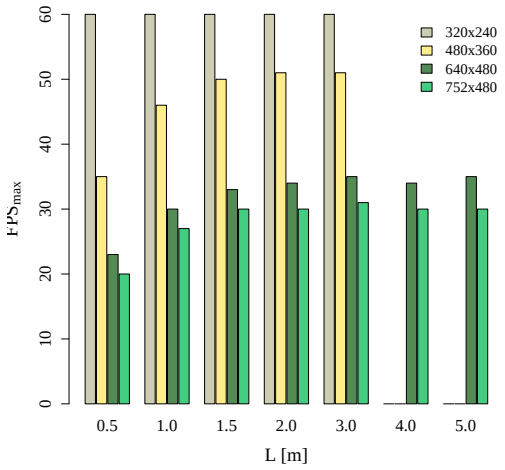

Fig. 2. Maximal achieved FPS.

The maximal achieved FPS using the tracking is shown in Fig. 2 for up to 5 m far blob. For the lowest resolution considered, the processing is limited by the Caspa camera, which can provide images at 60 Hz. On the other hand, a lower resolution limits the maximal measured distance, because of minimal required pixels in (1). In the case of the 320×240 resolution, a blob is reliably detected for $L \leq$3.2 m. For L=3.5 m, the blob is detected approximately in about 4 % cases. For the 480×360 resolution L_{max}=3.5 m, and for L=3.8 m the blob is detected in about 15 % cases. For longer distances the blob is not detected reliably.

B. Error of the Relative Distance Estimation

In this evaluation of the real expected error of the provided estimation of the relative distance, we consider the similar setup as for the previous evaluation, i.e., the pattern is placed at the camera optical axis, the error is measured using a client computer receiving information about the detected blobs from the camera module and comparison of the provided distance with the ground truth measured manually.

We aim to provide information about real achievable precision of the relative localization for the end-users, therefore, we consider the known camera intrinsic parameters. However, we expect that the camera model is not perfectly identified (or all parameters precisely estimated), e.g., because of imprecise calibration of the camera and other hard to estimate factors like the camera sensor charge leakage. Therefore, we assume a systematic error proportional to the measured distance and identify the systematic error using the real distances and the Least Square Method (LMS). The corrected average distance using the identified systematic

error is denoted as $\hat{L}$, and expected error in the distance estimate is computed as $L_e = |L - \hat{L}|$, where L is the ground truth. The standard deviation is considered as the repeatability of the measurements and it is presented in percents of the measured distance and it is denoted as L_δ.

TABLE III
ERROR AND RELIABILITY OF DISTANCE MEASUREMENTS IN x-AXIS

L	**320×240**		**480×360**		**640×480**		**752×480**	
	L_e	L_δ	L_e	L_δ	L_e	L_δ	L_e	L_δ
[m]	[cm]	[%]	[cm]	[%]	[cm]	[%]	[cm]	[%]
0.5	1.2	0.1	0.9	0.4	3.6	0.6	4.3	1.2
1.0	0.1	0.1	0.3	0.1	1.5	0.3	2.3	0.6
1.5	0.6	0.1	0.9	0.1	0.1	0.1	0.4	0.5
2.0	0.2	0.5	1.0	0.1	1.1	0.1	0.8	0.1
2.5	2.0	0.2	0.0	0.2	0.7	0.2	1.8	0.2
3.0	1.2	0.2	0.7	0.3	0.0	0.2	4.0	0.2
3.2	3.0	0.2	1.8	0.7	3.5	0.2	2.3	0.2
3.5	-	-	1.8	0.9	0.8	0.2	2.2	0.2
4.0	-	-	-	-	5.4	0.4	3.3	0.4
4.5	-	-	-	-	2.7	0.3	2.5	0.2
5.0	-	-	-	-	2.4	0.6	3.4	0.6
5.5	-	-	-	-	6.6	0.5	6.5	0.7

The provided information about the detected blob consists of its 3D coordinates x, y, z, estimated spatial orientation, and two pixel ratios of the blob (ratio of dark and light pixels and ratio of the current and expected number of pixels). Although orientation is estimated, only the position coordinates are considered in the experimental evaluation of the relative localization system presented here. The coordinate system used in the evaluation is schematically depicted in Fig. 1.

The systematic error has been identified for the pattern placed at the camera optical axis (x-axis) and for each considered resolution. The expected errors for the measured L are depicted in Table III. The results indicate that a higher resolution provides more precise estimations at longer distances, but principally it provides the same performance. Regarding the standard deviations, the repeatability of the measurements are in units of millimeters, which means tenths of the percentage points of the measured distance. A summarized expected error and reliability is shown in Table IV, where maximal values of L_e and L_δ over all evaluated distances L are presented, L_{max} is the maximal distance for a reliable blob detection.

TABLE IV
EXPECTED ERROR AND RELIABILITY OF DISTANCE ESTIMATION

Resolution	L_{max} [m]	L_e [cm]	L_δ [%]
320×240	3.2	3.0	0.5
480×360	3.5	1.8	0.7
640×480	5.5	6.6	0.6
752×480	5.5	6.5	0.6

C. Operational Space

The determination of the real operational space has been verified only for the resolution 480×360, because it provides a good trade-off between the achieved FPS and the maximal measured distance. In this experimental setup, the camera has been facing a 3 m distant wall and the pattern has been placed into various locations on the wall providing measurements in y and z axes.

We found out that at the 3 m distance the usable field of view is about 4 m×2 m, which corresponds with the values given by (2). However, the model introduced in Section V-C does not take into account the radial distortion. Typically, the barrel distortion causes the operational space to be larger than expected. If a pattern is placed at the y=0 m, it is detectable up to the vertical position z=0.9 m (the theoretical value is 0.96 m), where $(y = 0, z = 0)$ is at the camera axis. But, it is not the case for y=−0.5 m that allows to reliably detect the blob for z=1.0 m, or y=1 m providing maximal vertical position of the blob z=1.1 m.

D. Detectability of the Pattern

The aforementioned results have been obtained using a pattern with d=18 cm. It is clear that the size of the pattern affects practical deployment of the presented localization system. Small patterns can be more practical; however, the size of the pattern affects the maximal measurable distance as too small pattern can be hardly detected from long distances. Four patterns of different sizes have been considered and the maximal distances of the blob detection are presented in Table V. Notice, how the dimensions of the bounding circle (d=7.5 cm and d=5.5 cm) affect the detectability. A wider blank space around the disc increases detectability a bit, in particular for the 480×360 resolution. Regarding the results a smaller pattern can be used; however, with limited maximal distance between the pattern and the camera module.

TABLE V
MAXIMAL MEASURABLE DISTANCE FOR PATTERNS OF DIFFERENT SIZE

Pattern parameters (d_0, d_i, d) [cm]	A	B	C	D
Resolution	L_{max}	L_{max}	L_{max}	L_{max}
320×240	0.5	0.5	1.0	1.0
480×360	0.5	1.0	1.5	2.0
640×480	1.0	1.2	2.5	2.5
752×480	1.4	1.5	2.5	2.5

A=$(3.5, 2.1, 5.5)$, B= $(3.5, 2.1, 7.5)$, C=$(6.0, 3.6, 8.0)$, D=$(7.0, 4.2, 9.0)$, L_{max} in meters

VI. PRACTICAL DEMONSTRATION OF THE SYSTEM DEPLOYMENT

The developed localization system has been deployed in two scenarios for demonstration of its practical usability. First, it has been considered for the relative localization of a flying MAV operating in an outdoor environment. Although the MAV's movements are relatively fast, we did not observe issues with the blurred images and the system detected the pattern reliably. An example of detected pattern attached to a flying MAV is depicted in Fig. 3.

The second demonstration is shown in Fig. 4 that is aimed to provide an overview of the system performance in an indoor environment with varying lighting conditions.

Fig. 3. Detection of the pattern attached to a flying MAV.

Fig. 4. Demonstration of the relative localization system in a formation control: (top) images captured by the camera and the detected blob; (bottom) overview of the scene to show how the lighting conditions are changed.

VII. Conclusion

This paper presents description and real achievable performance characteristics of a simple relative localization system based on off-the-shelf components. The principle of the blob detection is based on well known standard approaches that have been implemented for a relatively cheap device available on the market. Even though the localization is based on an explicit pattern, the system provides sufficient precision for a robotic (and especially multi-robotic) tasks both in indoor and outdoor environments with changing lighting conditions.

The presented system can be considered as an enabling technology for practical experimenting and verification of novel approaches that are currently limited to laboratory or even to simulation only. The module is light-weight that allows to be attached to micro aerial vehicles. It enables to study swarm behaviors of real flying robots, which in fact is the main motivation for developing the system. The system itself together with the presented models is suited for real-world deployment of swarms of MAVs and it may be employed as a building block for a further investigation of novel swarm stabilization and planning strategies. The software is available for free and it works on standard hardware that is its advantage over other similar solutions.

The described localization system currently provides estimation of a single detected object only; however, the blob detection algorithm principally finds all the segments. Thus, it can be modified to provide estimation of relative position of several blobs, which in consequence can be used for a global localization system. Although the precision of such a system (regarding the results presented) can be expected in units of centimeters, the main benefits of the system would be its low-cost and its easy deployment in an outdoor environment. Developing such a complex localization system is a subject of our future work.

Acknowledgments

This work has been supported by the Ministry of Education of the Czech Republic under Projects No. LH11053 and 216240, by Czech Science Foundation GAČR under post-doctoral research projects No. P103/13/18316P and P103/12/P756 and by EU project No. 7E08006.

References

[1] J. Faigl, T. Krajník, V. Vonásek, and L. Přeučil, "On Localization Uncertainty in an Autonomous Inspection," in *ICRA*, 2012, pp. 1119–1124.

[2] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[3] W. Cai, Q. Yu, and H. Wang, "A fast contour-based approach to circle and ellipse detection," in *WCICA*, 2004, pp. 4686–4690.

[4] A. A. Rad, K. Faez, and N. Qaragozlou, "Fast circle detection using gradient pair vectors," in *DICTA*, 2003, pp. 879–888.

[5] Y. Xie and Q. Ji, "A new efficient ellipse detection method," in *ICPR (2)*, 2002, pp. 957–960.

[6] L.-Q. Jia, H.-M. Liu, Z.-H. Wang, and H. Chen, "An effective non-HT circle detection for centers and radii," in *ICMLC*, 2011, pp. 814–818.

[7] G. Chen, J. Ji, and L. Sun, "A novel circle detector based on sub-image point pairs," in *ICIA*, 2008, pp. 661–666.

[8] M. Carreras, P. Ridao, R. García, and T. Nicosevici, "Vision-based localization of an underwater robot in a structured environment," in *ICRA*, 2003, pp. 971–976.

[9] S. Lange, N. Sunderhauf, and P. Protzel, "A vision based onboard approach for landing and position control of an autonomous multirotor uav in gps-denied environments," in *ICRA*, 2009, pp. 1–6.

[10] L. Garca Carrillo, E. Rondon, A. Sanchez, A. Dzul, and R. Lozano, "Stabilization and trajectory tracking of a quad-rotor using vision," *Journal of Intelligent & Robotic Systems*, vol. 61, pp. 103–118, 2011.

[11] M. Bošnak, D. Matko, and S. Blažič, "Quadrocopter control using an on-board video system with off-board processing," *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 657–667, Apr. 2012.

[12] A. Masselli and A. Zell, "A Novel Marker Based Tracking Method for Position and Attitude Control of MAVs," in *Proceedings of International Micro Air Vehicle Conference and Flight Competition*, Braunschweig, Germany, 2012, pp. 1–6.

[13] S. Yang, S. A. Scherer, and A. Zell, "An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle," in *ICUAS*, 2012.

[14] A. W. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least-squares fitting of ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476–480, may 1999.

[15] G. Jiang and L. Quan, "Detection of concentric circles for camera calibration," in *IEEE ICCV*, 2005, pp. 333–340.