IEEE Copyright Notice

This is a submitted version of the paper:

Krajník, et al.: *External Localization System for Mobile Robotics*, In International Conference on Advanced Robotics, Montevideo, 2013.

The full version of the article is available on IEEE Xplore or on request.
For questions or requests, email Tom Krajnik or visit author's homepage.

External Localization System for Mobile Robotics

Tomáš Krajník
Lincoln Centre for Autonomous Systems
Univ. of Lincoln, United Kingdom
tkrajnik@lincoln.ac.uk

Matías Nitsche
Lab. of Robotics and Emb. Systems
Univ. of Buenos Aires, Argentina
mnitsche@dc.uba.ar

Jan Faigl
Dpt. of Computer Science and Engineering
FEE, Czech Technical Univ. in Prague
faiglj@fel.cvut.cz

Tom Duckett
Lincoln Centre for Autonomous Systems
Univ. of Lincoln, United Kingdom
tduckett@lincoln.ac.uk

Marta Mejail
Lab. of Robotics and Emb. Systems
Univ. of Buenos Aires, Argentina
marta@dc.uba.ar

Libor Přeučil
Intelligent and Mobile Robotics Group
FEE, Czech Technical Univ. in Prague
preucil@labe.felk.cvut.cz

***Abstract*—We present a fast and precise vision-based software intended for multiple robot localization. The core component of the proposed localization system is an efficient method for black and white circular pattern detection. The method is robust to variable lighting conditions, achieves sub-pixel precision, and its computational complexity is independent of the processed image size. With off-the-shelf computational equipment and low-cost camera, its core algorithm is able to process hundreds of images per second while tracking hundreds of objects with millimeter precision. We propose a mathematical model of the method that allows to calculate its precision, area of coverage, and processing speed from the camera's intrinsic parameters and hardware's processing capacity. The correctness of the presented model and performance of the algorithm in real-world conditions are verified in several experiments. Apart from the method description, we also publish its source code; so, it can be used as an enabling technology for various mobile robotics problems.**

I. Introduction

Although internal localization methods such as SLAM based approaches [1] are fundamental for autonomous robots, an external positioning reference (i.e., the ground truth) without any cumulative error is a key component for a proper evaluation of their performance. Moreover, precise external localization systems are also enabling technology for a wide field of possible applications, where some sort of additional supporting infrastructure can be used. Regarding this, the most known external localization reference is GPS; however, it is also known that it cannot be used indoors due to signal unavailability and also its precision may not be sufficient for navigation of small vehicles. These limitations motivate a design of several localization principles, which can be broadly divided into two major groups by means of the type of sensors used: active and passive.

In the former group, technologies such as LED emitters [2], radio beacons, and ceiling projection [3] are used, among others. However, a widely used approach in recent works [4], is the commercial motion capture system from ViCon [5], which uses high-resolution and high-speed IR (infra-red) cameras, IR emitters and reflective targets. Even though this system achieves great performance and accuracy, it is a costly solution not applicable to every research environment.

On the other hand, several passive vision-based localization methods have been proposed in literature. Most of these approaches employ augmented-reality oriented markers such as ARTag [6] and ARToolKit+ [7], which not only allow obtaining the pose of the target but can also encode some additional information. These target detectors were used in several works in order to obtain pose information of mobile robots [8], [9], [10], [11]. Alternative target shapes are also proposed in recent literature, which are specifically designed for vision-based localization systems with a higher precision and reduced computational costs. Due to several positive aspects, circular shaped patterns appear to be best suited as fiducial markers in external localization systems and can be found in several works [12], [13], [14], [15], [16].

The herein presented vision-based external localization system is based on black and white circular planar ring pattern (roundel) detector. The idea of the proposed pattern detection has been originally motivated by practical needs for a relative localization of small unmanned aerial vehicles (UAV) [17] to keep them in a swarm formation where a fast image processing using only on-board computational resources is required. Considering the previous passive localization approaches, we found out that the proposed roundel detector best fits our needs for fast estimation of a relative distance of flying UAVs using only the on-board computational resources.

In this paper, we consider the principle of the roundel detection [17] in a global localization scenario, where the method is able to detect and track several thousands of images per second on a common desktop computer. Moreover, we propose 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. The model provides estimation of coverage and precision of the localization system from the resolution of the camera and size of the patterns, which allows a user to choose between high-end and low-end cameras depending on the specific application.

Low computational requirements are based on exploiting the fact the algorithm allows to initiate the pattern search anywhere in the image without any performance penalty. Thus, the search for a pattern is started from the point of the last known pattern position. Since the algorithm does not contain any phase that processes the entire image, successful tracking causes the method to process only the area occupied by the pattern, which makes the algorithm's computational complexity independent of the image size.

978-1-4799-2722-7/13/$31.00

In addition, the system is easy to use and allows a fast deployment, which is a useful feature for considering it as an enabling technology for various robotic projects. The deployment of the system does not require user-set parameters or intricate setup process and user interaction with the external localization system application is reduced to minimum. Only an initial unattended calibration step is needed to define the reference frame and to compute a pose of ground robots moving on a plane with a millimeter precision using an off-the-shelf camera.

The system has been already deployed in a number of international mobile robotics projects concerning distributed quad rotor localization [17], visual based autonomous navigation [18], decentralized formation control [19], evolutionary swarm [20] and educational [15] robotics. Since the system has already proved to be useful in a variety of applications, we publish its source code [21]; so, other roboteers can use it for their projects.

II. Method Description

The proposed system is based on a fast and precise detection of a black and white roundel, which consists of two concentric annuli with a white central disc. An initial detection step is performed to identify the position of the roundel in pixel coordinates. Using camera re-projection techniques and known dimensions of the inner and outer roundels, the three-dimensional position of the target with respect to the camera is computed. Finally, a transformation of the coordinate frame is applied to compute target coordinates with respect to a user-defined frame, either in three or two dimensions, depending on the chosen scenario.

A. Pattern detection

The pattern detection itself is based on an on-demand thresholding and a flood fill technique, which computes the bounding box, number of pixels, and center of gravity on the fly. Initially, when there is no *a priori* information about the target position in the image, the detection seeks possible candidates and consecutively tests them for certain properties (with increasing complexity), which determines a positive match. In subsequent detection steps, when a prior target position is available, the algorithm starts detection over this area. If the target is successfully re-detected (i.e., it is tracked), the detection step involves only pixels belonging to the target itself. Since the method is quite robust (see the following sections for detection limits), tracking is generally successful resulting in a very high performance detection.

1) Segmentation and thresholding: Starting from a pixel position p_0, the algorithm classifies the current pixel as either black or white by thresholding against an adaptively set value. In a case the pixel is classified as white, the algorithm proceeds to the next image pixel. In case a black pixel is found, a segment consisting of the contiguous set of black pixels is computed using a queue-based flood-fill method, see Algorithm 1. This black segment is tested for a possible match of the outer ring of the pattern. At this point, these tests consist of a minimum size (in terms of the number of pixels belonging to the segment) and a roundness measure within acceptable bounds. These simple constraints allow a fast rejection of false positives. In a case either test fails, the detection for further black segments continues starting from the next pixel position. However, no redundant computation is performed since the previous segment is labeled with a unique identifier.

Algorithm 1: examineCircle

Input: (p, ρ_{exp}, c): p - starting pixel position, ρ_{exp} - expected area to bounding box dimensions ratio, c - searched segment type (white or black)
Output: $(u, v, v_x, v_y, \mu, \texttt{valid})$: (u, v) - segment center, (v_x, v_y) - bounding box, μ - average brightness, `valid` - validity

$s_{id} \leftarrow s_{id} + 1$ `// increment segment ID`
`pixel_class`[p] $\leftarrow s_{id}$
`queue`[q_{end}++]$\leftarrow p$
$q_{old} \leftarrow q_{end}$
while $q_{end} > q_{start}$ **do**
  $q \leftarrow$ `queue`[q_{start}++]
  foreach *offset* $\in \{+1, -1, +w, -w\}$ **do**
    $r \leftarrow q +$ offset
    if *`pixel_class[r]` = 0* **then**
      `pixel_class`[r] $\leftarrow$ classify(r)
    if *`pixel_class[r]`* $= c$ **then**
      `queue`[q_{end}++]$\leftarrow r$
      `pixel_class`[r] $\leftarrow s_{id}$
      update x_{min},x_{max},y_{min},y_{max} from r_x,r_y
$s \leftarrow q_{end} - q_{oldend}$
if $s > min\_size$ **then**
  $u, v \leftarrow (x_{max} + x_{min})/2, (y_{max} + y_{min})/2$
  $v_x, v_y \leftarrow (x_{max} - x_{min}) + 1, (y_{max} - y_{min}) + 1$
  $\rho \leftarrow \rho_{exp} v_x v_y / s - 1$ `// roundness`
  if $-\rho_{tol} < \rho < \rho_{tol}$ **then**
    $\mu \leftarrow \frac{1}{s} \sum_{j=q_{start}}^{q_{end}-1}$ `Image[j]`
    `valid` $\leftarrow$ true

The number of pixels s of an elliptic ring with outer and inner diameters d_o, d_i, respectively, and dimensions b_x, b_y should be

$$s = b_x b_y \pi \frac{d_o^2 - d_i^2}{d_o^2}. \quad (1)$$

Thus, the roundness measure ρ can be calculated as:

$$\rho = \frac{b_x b_y \rho_{pred}}{s}, \quad (2)$$

where ρ_{pred} can be estimated from d_o and d_i. The roundness test accepts a segment if ρ is within a tolerance range given by pattern deformation and spatial orientation.

If a black segment passes both tests, the detection resumes from the pixel position corresponding to the segment centroid where a white pixel is expected. In this case, the corresponding white segment is computed using the flood-fill algorithm. If the minimum size and roundness tests are valid, further validation tests are performed involving the centroid positions of both segments, their area ratio, and a more complex circularity measure (discussed in following sections). If the segments pass all these tests, the pattern is considered to be found and its centroid position will be used as a starting point p_0 for the next detection run. This detection method is described in detail in Algorithm 2.

Algorithm 2: detectCircle

Input: $(p_0, \tau,$ `Image`$)$: p_0 - position to start the search, τ - threshold, `Image` being processed
Output: (c, p_0, τ): c - the pattern data, p_0 - position to start the next search, τ - an updated threshold

```
if first run then
    size ← sizeof(Image);
    initialize pixel_class[size]
    initialize queue[max_queue_size]
s_id ← 0
i ← p_0
repeat
    if pixel_class[i] = unknown then
        if classify(i,τ) = black then
            pixel_class[i] ← black
    // initiate pattern search
    if pixel_class[i] = black then
        // search for outer ring
        q_end ← q_start ← 0
        c_outer ← examineCircle(i,ρ_outer,black)
        if c_outer is valid then
            // search for inner ellipse
            j ← center of (c_out)
            c_inner ← examineCircle(j,ρ_inner,white)
            if c_inner is valid then
                compute c_outer and c_inner centers
                check for concentricity
                compute ellipse semiaxes e_0, e_1
                if q_end = π|e_0 e_1| then
                    mark segment as valid
                    break
    i ← (i + 1 mod size)    // go to next pixel
until i ≠ p_0;
// hide pattern from other detectors
if c_inner is valid then
    paint over all inner ellipse pixels as black
    τ ← average brightness of c_outer and c_inner
else
    τ ← pseudo-random number
// cleanup
if only two segments examined then
    reset pixel_class[] inside bounding box of c_outer
else
    reset all pixels in pixel_class[] array
```

2) Efficient thresholding: Thresholding is performed on-demand over each pixel analyzed during the detection. For the first access, the RGB values are read and the particular pixel is classified as either black or white. In the subsequent steps, the classification result for this pixel is re-used. In cases when tracking is successful, only the relevant pixels are thresholded, and therefore, the detection step is not directly dependent on the input image resolution. Clearing the per-pixel classification memory area is also efficiently performed by only resetting values inside the roundel bounding box. When the detection fails, extra memory accesses resulting from this strategy are negligible compared to a full-image thresholding approach.

If the detection of a roundel fails, the threshold parameter τ is adaptively updated by a simple binary search scheme over the range of possible values, up to a pre-defined granularity level, when τ is reset to the initial value. Otherwise, if the detection is successful, the threshold is updated by using the information obtained during the detection in order to iteratively improve the precision of the segmentation step:

$$\tau \leftarrow \frac{(\mu_o + \mu_i)}{2}, \tag{3}$$

where μ_o, μ_i correspond to the mean brightness value of the outer and inner segments, respectively.

3) Pattern center and dimensions: Once a pattern passes all the aforementioned tests, the queue, which has been used by the flood-fill phase, contains positions of all the pattern pixels. Thus, all the information to calculate the ellipse centre u, v and semiaxes $\mathbf{e_0}, \mathbf{e_1}$ is at hand. The pattern center u, v is calculated simply as the mean of the pixel positions. Then, the method calculates the covariance matrix of the pixels positions $\mathbf{C}$. The eigenvalues and eigenvectors of the matrix correspond to the ellipse semiaxes lengths and orientations. Finally, the segment circularity is verified by checking if $n \approx \pi|\mathbf{e_0}||\mathbf{e_1}|$.

Since the matrix $\mathbf{C}$ is two-dimensional, its eigen decomposition is a matter of solving one quadratic equation. Note that if it is desirable, the most of the calculation of $\mathbf{C}$ can be performed in the integer arithmetic, which might be useful if the system runs on an embedded hardware.

4) Multiple target detection: So far, the detection of a single pattern was considered. However, the proposed localization system is capable of detecting and tracking multiple patterns simultaneously, simply by running the detector several times consecutively. Each detector changes the color of the inner circle pixels to black to prevent multiple detection of the same target. Besides, each detector also uses a different threshold τ, which increases the system robustness to uneven illumination across the working area.

B. Pattern localization

The pattern position relative to the camera is determined from the parameters calculated at the previous step. We assume that the radial distortion of the camera is not extreme and its intrinsic parameters have been established by the method [22] or similar. Then, the ellipse center and semiaxes are calculated from the covariance matrix eigenvectors and transformed to a canonical camera coordinate system. The transformed parameters are then used to establish coefficients of the ellipse characteristic equation. These coefficients are subsequently utilized to calculate the pattern's spatial orientation and its position within the camera coordinate frame.

A canonical form refers to a pinhole camera model with unit focal lengths and no radial distortion. The transformation to a canonical camera system is basically an inverse transform to the transformation of spatial coordinate to image coordinates. To transform the ellipse center and semiaxes to canonical camera coordinates, we calculate the ellipse (co)vertices, transform them to the canonical camera coordinates, and use the transformed vertices to calculate the axes and center again. The canonical center and axes are then used in the ellipse characteristic equation, which allows to determine its 3D position and orientation [13].

C. Transformation to global coordinates

The position $\mathbf{x_c}$ of the circular pattern established in the previous step is in a camera centered coordinate frame. Depending on the particular application scenario, our system allows to transform the pattern coordinates to 3D and 2D coordinate frames defined by the user. The user just places three (to define a 3D coordinate frame) or four (to define a 2D coordinate frame) circular patterns in the coordinate frame and provides the system with their real positions.

1) Global coordinate frame – 3D case: In the case of the 3D localization, the three patterns at positions $\mathbf{x_0}, \mathbf{x_1}, \mathbf{x_2}$ define coordinate origin and x and y axes, respectively. The transformation between the global $\mathbf{x} = (x, y, z)^T$ and camera centered $\mathbf{x_c} = (x_c, y_c, z_c)^T$ coordinate system can be represented as $\mathbf{x} = \mathbf{T}(\mathbf{x_c} - \mathbf{x_0})$, where the matrix $\mathbf{T} = (\mathbf{t_1}, \mathbf{t_2}, \mathbf{t_3})^{-1}$ represents a transformation of the vector $\mathbf{x'} = \mathbf{x_c} - \mathbf{x_0}$ to the coordinate system defined by the orthonormal basis $\mathbf{t_1}, \mathbf{t_2}, \mathbf{t_3}$, where $\mathbf{t_1} = \mathbf{x_1} - \mathbf{x_0}$, $\mathbf{t_2} = \mathbf{x_2} - \mathbf{x_0}$ and $\mathbf{t_3} = \mathbf{t_1} \times \mathbf{t_2}$.

2) Global coordinate frame – 2D case: The precision of the 3D localization depends mainly on the precision of the pattern distance estimation, which is generally low. This source of localization error can be avoided for patterns that are located only on a plane, e.g., in a case of localization of ground robots operating on a floor. Here, the transformation from the image coordinates to an arbitrary world plane is a homography, and (homogeneous) spatial coordinates $\mathbf{x}$ of the patterns can be calculated directly from their canonical coordinates $\mathbf{u'}$ simply by $\mathbf{x} = \mathbf{H}\mathbf{u'}$, where $\mathbf{H}$ is a 3×3 homography matrix. Similarly to the case of three-dimensional localization, the user can define $\mathbf{H}$ just by placing four patterns in the camera field of view and providing the system with their positions in the desired coordinate frame.

III. Estimating the system performance

In this section, we present a set of equations that can be used to estimate the coverage, precision, and processing speed of the system. Using these equations, the user can choose an appropriate camera, computational resources, and pattern size for a particular application.

A. Localization system coverage

An important property of the system is its "coverage", i.e., the space where the pattern is reliably detected and localized. System coverage has a pyramidal shape with its apex at the camera, see Figure 1.

We denote the pattern's maximal detectable distance as v_x and the base dimensions as v_y and v_z. The circular pattern appears in the camera image as an ellipse. In order to be detected reliably, the minor ellipse axis of the projected pattern must exceed a critical value, which we define as D. The length (in pixels) of the minor ellipse semiaxis e_1 can be estimated by

$$e_1 = \frac{1}{2} f_c \frac{d_o}{x} \cos(\varphi), \tag{4}$$

where x is the pattern distance from the image plane, d_o is the pattern diameter, and φ represents the pattern tilt. Rewriting

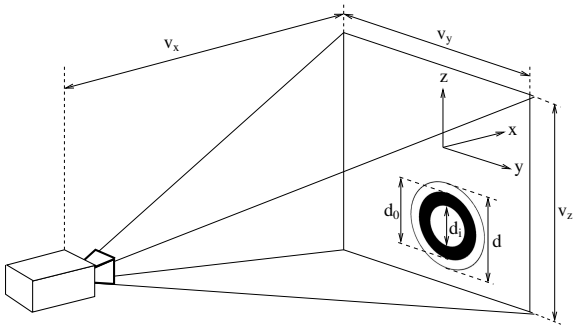

Fig. 1: Geometry of the operational space.

equation (4), the maximal detectable distance v_x of the pattern can be calculated as

$$v_x \quad = \quad f_c \frac{d_o}{D} \cos(\varphi). \tag{5}$$

The value of D has been experimentally established as 14 pixels. Notice that increasing camera resolution while maintaining its field of view increases the focal length f_c; so, setting the camera resolution as high as possible maximizes the system coverage.

Knowing the value of v_x, the base dimensions v_y and v_z can be established as

$$v_y = v_x w f_c^{-1}, \qquad v_z = v_x h f_c^{-1}, \tag{6}$$

where w and h are the horizontal and vertical resolutions of the camera, respectively.

B. Localization system precision

Another important property of the localization system is its precision. The localization error is different for the three-dimensional and two-dimensional system modes.

In the case of the full 3D localization, the main source of the localization imprecision is an incorrect estimation of the pattern distance. Since the pattern distance is inversely proportional to its diameter in pixels, smaller patterns will be localized less precisely. The expected relative localization error can be roughly estimated as

$$\eta_{3D} = \eta_f + \eta_d = \Delta_f f_c^{-1} + \Delta_n e_0^{-1}, \tag{7}$$

where Δ_f is the error of the focal length estimation, Δ_n represents an error of the ellipse axis estimation due to image noise and e_0 is the pattern's major semiaxis length in pixels. While Δ_n has to be established experimentally, Δf and f_c are known camera parameters. For a well-calibrated camera, $\Delta_f f_c^{-1}$ is small and thus the major source of distance estimation error is the ratio of image noise to the pattern image size. This suggests that the precision of the 3D localization can be simply increased by using a higher resolution camera or a larger pattern.

In the two-dimensional localization mode, the pattern position is estimated simply from its centre image coordinates by means of homography. Assuming that the homography is established correctly, the localization error is affected mainly by the image radial distortion. Due to this fact, the system precision depends on the camera lens rather than on the camera resolution. A thorough model of radial distortion effect is beyond the scope of this paper.

C. Computational requirements

The computational time needed to process a single image can be estimated from the number of patterns n, their average size s_p, image size s_i, tracking failure rate α, and the processor (single core) speed m by a linear function:

$$t = (k_0 + k_1(s_p(1-\alpha) + s_i\alpha))\, nm^{-1}, \tag{8}$$

where k_1 is a constant corresponding to the number of computer operations per pixel (of the pattern), and k_0 represents the number of operations needed per pattern regardless of the pattern size. The values of k_1 and k_0 are estimated as 900 and 5.10^5, respectively. Having this model, we can estimate that processing speed for tracking 50 patterns with 30 pixel diameter on a dual core 53000 GIPS machine should take 1.2 ms, which allows to process 800 frames per second.

IV. Experiments

The model of the localization system presented in the previous sections has been experimentally validated in a series of practical deployments. First, a coverage of the robots' operational space has been identified with regards to the maximal distance of the pattern for a reliable detection. Then, the localization precision has been measured for both cases: 3D and 2D localization, see Section IV-B. Finally, real computational requirements are reported in Section IV-C.

A. Localization system coverage

The purpose of this experiment is to test the area coverage of the localization system. A crucial parameter of the coverage is the maximal distance of the reliable pattern detection v_x. The verification has been performed with two different cameras tracking three patterns with different diameters. Two sequences of images with increasing distance of the camera from the patterns were captured. While the first sequence has the patterns directly facing the camera, the second sequence has been taken from the 40 degree angle.

TABLE I: Maximal distance of pattern detection

Camera type	Pattern $d_o[cm]$	Distance [m] *Measured* 0°	*Measured* 40°	*Predicted* 0°	*Predicted* 40°
Logitech QC Pro	2.5	1.3	1.2	1.4	1.1
	5.0	3.1	2.6	2.8	2.1
	7.5	4.1	3.7	4.2	3.2
Olympus VR-340	2.5	6.5	5.9	6.7	5.1
	5.0	12.8	10.9	13.4	10.3
	7.5	19.3	17.1	20.1	15.4

The base dimensions v_y and v_z can be calculated from v_x and the camera parameters. To check whether Equation (6) is correct, we placed two patterns at the opposite corners of the image and compared their positions with the base dimensions. The measured and predicted distances are presented in Table I.

B. Localization precision

The system localization precision has been evaluated for 20 patterns placed on the floor of a large room and by a manual measuring their precise positions. Then, we took several pictures of the scene with three different cameras from two different viewpoints ("top" and "side"). The cameras used were: Creative Live! webcam, Olympus VR-340, and Canon 550D that has been set to 1280×720, 4608×3456, and 5184×3456 pixel resolutions, respectively.

Pre-placed roundels were used to define global coordinates as described in Section II-C. An average distance of the known pattern positions to the ones estimated by the system was considered as a measure of the localization error. Both 2D and 3D localization errors are summarized in Table II and Table III, which also contain a predicted average localization error (ϵ_{pred}) calculated by (7). The results show that the

TABLE II: Precision of 3D position estimation

Image camera	view	Abs. [cm] ϵ_{avg}	ϵ_{max}	Rel. [%] η_{pred}	η_{avg}	η_{max}
Webcam	side	7.5	17.8	1.01	1.13	2.67
Webcam	top	3.9	11.1	0.73	0.59	1.67
VR-340	side	1.9	6.2	0.28	0.28	0.90
VR-340	top	2.7	7.1	0.26	0.31	0.82
C-550D	top	2.0	5.1	0.19	0.34	0.86

TABLE III: Precision of 2D position estimation

Image camera	view	Abs. [cm] ϵ_{avg}	ϵ_{max}	Rel. [%] η_{avg}	η_{max}
Webcam	side	0.23	0.62	0.05	0.12
Webcam	top	0.18	0.68	0.04	0.13
VR-340	side	0.64	1.40	0.12	0.26
VR-340	top	0.68	2.08	0.10	0.28
C-550D	top	0.15	0.33	0.04	0.08

position estimation of the 2D localization is more precise than the 3D one. The measured values also confirm the assumption that while the 3D localization precision increases with the camera resolution, the 2D localization precision does not.

C. Computational requirements

Equation (8) suggests that if the tracking works correctly, the computational requirements of the method are independent of the image size and it is affected only by the number and size of the tracked patterns. Three datasets have been created to verify this assumption. The first dataset consists of a fixed number of patterns of uniform size, but the dimensions of the dataset images vary. The second dataset consists of uniformly sized images with a variable pattern size. The third dataset has both image and pattern dimensions fixed, but the number of patterns increases from zero to two hundreds. The dataset images were processed by a single core Intel i5 CPU running at 2.5 GHz. The average processing times to search a single image for all the required patterns are shown in Figure 2.

The results clearly prove that in the case of perfect tracking, the image processing time is proportional to the number of pixels occupied by the patterns and is unaffected by the image dimensions. Moreover, the results demonstrate a good scalability of the algorithm, which can track two hundred targets at more than one hundred times per second.

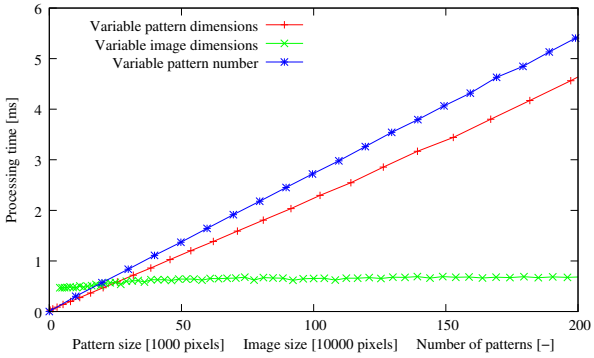

Fig. 2: Influence of the number of tracked patterns, pattern size, and image size on the computation time.

TABLE IV: The detection speed using individual platforms

CPU	Processing time [ms]			
	Measured		*Predicted*	
	Static	Moving	Static	Moving
i5 2450M	0.04	0.37	0.04	0.35
Atom N270	0.30	3.25	0.33	2.68
Pentium M	0.20	1.44	0.17	1.45

In addition, we tested the algorithm on three different platforms and two additional real-world datasets to further verify the soundness of (8). The first dataset consists of images of a static, 700 pixel pattern; and the second dataset contains 130 images of a fast moving pattern, which becomes obstructed in one of the frames, causing the tracking to fail. The average computational times to process one image of each dataset (according to (8)) are summarized in Table IV. The results indicate the correctness of the model described in Section III-C.

V. Conclusion

In this paper, we present a fast and precise vision-based system intended for multiple robot localization. Its core algorithm is based on a novel principle of circular roundel detection that has computational complexity independent of the processed image size. The resulting system allows to localize a large robotic swarm with a millimeter precision, while keeping up with standard camera frame rates. In addition, we propose a model of the localization system to aid its potential users in deciding what kind of equipment to use for their particular setup, The model allows to calculate the camera and computer parameters from the desired localization precision, coverage, and update rate. It is also worth to mention that the system is made of low-cost off-the-shelf components, since a camera and printable patterns are the only required elements.

In future, we plan to increase the precision and coverage of the system by using multiple cameras. We also aim to improve the tracking success rate by predicting the position of the target by considering the dynamics of the tracked object.

Acknowledgments

The work has been supported by the EU projects ICT-216240, ICT-600623 ‘STRANDS’ and Czech Science Foundation project 13-18316P.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.

[2] A. Breitenmoser, L. Kneip, and R. Siegwart, “A monocular vision-based system for 6D relative robot localization,” in *IROS*, 2011, pp. 79–85.

[3] Y. Yamamoto, P. Pirjanian, M. Munich, E. DiBernardo, L. Goncalves, J. Ostrowski, and N. Karlsson, “Optical sensing for robot perception and localization,” in *IEEE Workshop on Advanced Robotics and its Social Impacts*, 2005, pp. 14–17.

[4] D. Mellinger, N. Michael, and V. Kumar, “Trajectory generation and control for precise aggressive maneuvers with quadrotors,” *International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, Jan. 2012.

[5] Vicon, “Vicon MX Systems,” [cited 12 Jul 2013]. [Online]. Available: http://www.vicon.com/products/viconmx.html

[6] M. Fiala, “ARTag, An Improved Marker System Based on ARToolkit,” 2004.

[7] D. Wagner and D. Schmalstieg, “ARToolKitPlus for pose tracking on mobile devices,” in *12th Computer Vision Winter Workshop (CVWW)*, 2007, pp. 139–146.

[8] M. Fiala, “Vision guided control of multiple robots,” in *First Canadian Conference on Computer and Robot Vision*, 2004, pp. 241–246.

[9] I. Rekleitis, D. Meger, and G. Dudek, “Simultaneous planning, localization, and mapping in a camera sensor network,” *Robotics and Autonomous Systems*, vol. 54, no. 11, 2006.

[10] E. Stump, V. Kumar, B. Grocholsky, and P. M. Shiroma, “Control for Localization of Targets using Range-only Sensors,” *International Journal of Robotics Research*, vol. 28, no. 6, pp. 743–757, 2009.

[11] M. Bošnak, D. Matko, and S. Blažič, “Quadrocopter hovering using position-estimation information from inertial sensors and a high-delay video system,” *Journal of Intelligent & Robotic Systems*, vol. 67, no. 1, pp. 43–60, 2012.

[12] S. J. Ahn, W. Rauh, and M. Recknagel, “Circular coded landmark for optical 3d-measurement and robot vision,” in *IROS*, 1999, pp. 1128–1133.

[13] S. Yang, S. Scherer, and A. Zell, “An Onboard Monocular Vision System for Autonomous Takeoff, Hovering and Landing of a Micro Aerial Vehicle,” *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1-4, pp. 499–515, 2013.

[14] D. L. de Ipiña, P. R. S. Mendonça, and A. Hopper, “TRIP: A low-cost vision-based location system for ubiquitous computing,” *Personal and Ubiquitous Computing*, vol. 6, no. 3, pp. 206–219, 2002.

[15] M. Kulich, J. Chudoba, K. Košnar, T. Krajník, J. Faigl, and L. Přeučil, “Syrotek - distance teaching of mobile robotics,” *IEEE Trans. Education*, vol. 56, no. 1, pp. 18–23, 2013.

[16] S. Pedre, T. Krajník, E. Todorovich, and P. Borensztejn, “Hardware/software co-design for real time embedded image processing: A case study,” in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ser. Lecture Notes in Computer Science, L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, Eds. Springer Berlin Heidelberg, 2012, vol. 7441, pp. 599–606.

[17] J. Faigl, T. Krajník, J. Chudoba, L. Přeučil, and M. Saska, “Low-Cost Embedded System for Relative Localization in Robotic Swarms,” in *ICRA*, 2013, pp. 985–990.

[18] T. Krajník, M. Nitsche, S. Pedre, L. Přeučil, and M. Mejail, “A Simple Visual Navigation System for an UAV,” in *Int. Multi-Conf. on Systems, Signals and Devices*, 2012, pp. 34–34.

[19] M. Saska, T. Krajník, and L. Přeučil, “Cooperative μUAV-UGV Autonomous Indoor Surveillance,” in *Int. Multi-Conf. on Systems, Signals and Devices*, 2012, pp. 36–36.

[20] S. Kernbach, E. Meister, F. Schlachter, K. Jebens, M. Szymanski, J. Liedke, D. Laneri, L. Winkler, T. Schmickl, R. Thenius, P. Corradi, and L. Ricotti, “Symbiotic robot organisms: Replicator and symbrion projects,” in *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, 2008, pp. 62–69.

[21] T. Krajník, M. Nitsche, and J. Faigl, “The WhyCon system,” [cited 12 Jul 2013]. [Online]. Available: http://purl.org/robotics/whycon

[22] J. Heikkila and O. Silven, “A four-step camera calibration procedure with implicit image correction,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 1106–1112.