Metrol. Meas. Syst., Vol. 30 (2023) No. 3, pp. 423–439
DOI: 10.24425/mms.2023.146420

PAN POLSKA AKADEMIA NAUK

METROLOGY AND MEASUREMENT SYSTEMS
Index 330930, ISSN 0860-8229
www.metrology.wat.edu.pl

M&MS

HOMOGRAPHY AUGMENTED PARTICLE FILTER SLAM

Paweł Leszek Słowak, Piotr Kaniewski

Military University of Technology, Faculty of Electronics, Gen. S. Kaliskiego 2, 00-908 Warsaw, Poland
(✉ *pawel.slowak@wat.edu.pl*, *piotr.kaniewski@wat.edu.pl*)

Abstract

The article presents a comprehensive study of a visual-inertial simultaneous localization and mapping (SLAM) algorithm designed for aerial vehicles. The goal of the research is to propose an improvement to the particle filter SLAM system that allows for more accurate and robust navigation of unknown environments. The authors introduce a modification that utilizes a homography matrix decomposition calculated from the camera frame-to-frame relationships. This procedure aims to refine the particle filter proposal distribution of the estimated robot state. In addition, the authors implement a mechanism of calculating a homography matrix from robot displacement, which is utilized to eliminate outliers in the frame-to-frame feature detection procedure. The algorithm is evaluated using simulation and real-world datasets, and the results show that the proposed improvements make the algorithm more accurate and robust. Specifically, the use of homography matrix decomposition allows the algorithm to be more efficient, with a smaller number of particles, without sacrificing accuracy. Furthermore, the incorporation of robot displacement information helps improve the accuracy of the feature detection procedure, leading to more reliable and consistent results. The article concludes with a discussion of the implemented and tested SLAM solution, highlighting its strengths and limitations. Overall, the proposed algorithm is a promising approach for achieving accurate and robust autonomous navigation of unknown environments.

Keywords: Simultaneous Localization and Mapping (SLAM), homography matrix, particle filter, robot navigation, visual-inertial systems.

1. Introduction

Simultaneous localization and mapping (SLAM) is a technique for obtaining a trajectory of a robot together with a 3D structure of surroundings that the robot is navigating through. The purpose of research in this field is to develop an accurate and robust system consisting of a robotic platform and sensors to enable an autonomous vehicle to explore previously unknown environments. Achieving capability to perform SLAM is among the most promising and difficult challenges for unmanned platforms, as it would simplify other key tasks for future robots such as path planning, obstacle avoidance and object manipulation.

Article history: received March 31, 2023; revised May 19, 2023; accepted May 23, 2023; available online September 16, 2023.

Traditionally, two primary paradigms of SLAM are recognized. First, there are numerous filter approaches for simultaneous localization and mapping. Their main common trait is the procedure of estimating a multidimensional state vector, which comprises the vehicle pose together with a map of landmarks locations. One of the earliest attempts were built upon the *Extended Kalman Filter* (EKF) framework [1]. Other SLAM systems that fit in this filter category are different variants of *Kalman Filter* (KF) implementations – like Unscented KF and Information Filter – as well as systems incorporating a particle filter (PF) as a backbone of their architecture – where [2,3] and [4] are among the most notable examples. Another category of SLAM approaches are optimization-based solutions – with [5,6] and [7] as the most significant demonstrations. They are estimating the trajectory as a pose-graph structure, implementing *bundle adjustment* (BA) [8] where every node corresponds to a robot pose and they are connected by edges that model spatial constraints between the poses. The map is built using only selected camera frames – called keyframes.

It is important to note that SLAM systems implementing BA are frequently perceived to be more accurate than filter methods at the same computational cost, which was demonstrated in Strasdat *et al.* [9]. However, there are still some noticeable advantages of filter architecture systems, especially those based on PF. First, the PF SLAM approach can be considered a multiple hypotheses analysis [10] where every particle represents a different hypothesis concerning a robot pose and a map. The survival of the fittest approach to the resampling procedure can be described as a constantly running local relocalization procedure [11], contributing to the algorithm robustness. Further, the weighting procedure based on a likelihood multiplication and normalization [12] allows to integrate additional sensors (for example *ultrawideband* (UWB) radio transceivers [13]) with ease – simply as another multiplication factor. This makes the PF framework a potentially useful tool for multi-sensory platforms.

SLAM algorithms utilizing camera sensors are also frequently classified according to the type of the camera used. Pure visual SLAM approaches are based on a single monocular camera. Among the most known monocular algorithms not mentioned before are [14] and [15]. Robotic platforms performing SLAM can be further equipped with *inertial measurement units* (IMU) which are a useful additional source of motion information [16–18]. This type of SLAM architecture is known as a visual-inertial SLAM where [19, 20] and [21] are among those most prolific. Our approach also falls into the visual-inertial category. Algorithms which camera sensors able to measure depth are classified as RGB-D camera SLAM. Among the most advanced RBG-D SLAM algorithms are [22] and [23].

The purpose of the work presented in this paper was to expand our previously developed algorithm [24]. By introducing the proposal distribution refinement as in [10], we aimed at making the PF framework less computationally expensive through limiting the number of particles needed to accurately describe the *probability density function* (PDF) of the pose of the robot. Further, as the refinement is based on a homography matrix decomposition calculated from the frame-to-frame relationships, apart from the previously used frame-to-map relationship, it introduces a new information source making the filter more accurate and robust.

In this research we build on the main ideas and architecture of our previous SLAM system [24]. The authors' major contribution in this paper is threefold:

- implementation of a procedure that utilizes a homography matrix to refine proposal distribution of a PF,
- adding a mechanism that calculates a homography matrix from a robot displacement to eliminate outliers in frame-to-frame feature detection procedure,
- evaluation of the proposed improvements using simulation and real-world datasets.

Metrol. Meas. Syst., Vol. 30 (2023), No. 3, pp. 423–439
DOI: 10.24425/mms.2023.146420

The remainder of the paper is organized as follows. In Section 2, materials and methods are described in detail. In Section 3, the results of the SLAM algorithm are presented. Section 4 contains the conclusions where the approach and test results are summarised.

2. Materials and methodology

In this paper, we propose a modification of a particle filter SLAM algorithm that is an extension to the monocular SLAM approach detailed in [24]. Below, a review of the approach is given, as well as a detailed description of the proposed algorithm augmentation which exploits the epipolar geometry relationship between points extracted from subsequent frames captured by a camera during a UAV flight.

2.1. Particle filter SLAM approach

Our framework aims to solve the SLAM problem for an airborne autonomous platform in the event of absence of an external positioning signal from a geospatial positioning system. We assume that the UAV is equipped with an IMU and a downward facing gyro-stabilized monocular camera – criteria regularly met in aerial drones. According to the common classification adopted in numerous SLAM field surveys, including [25] and [26], our approach can be identified as a particle-filter-based visual-inertial monocular SLAM system.

The localization routine in our SLAM system is performed by estimating the UAV kinematic parameters $\mathbf{x}_k$ consisting of 9 variables:

$$\mathbf{x}_k = \begin{bmatrix} x & y & z & v_x & v_y & v_z & \phi & \theta & \psi \end{bmatrix}^T, \tag{1}$$

where k is a time step, x, y, z represent a localization in rectangular coordinates, v_x, v_y, v_z are orthogonal velocity components and ϕ, θ and ψ are roll, pitch and yaw orientation angles respectively. The state vector directly describes the camera pose, rather than the pose of a drone itself, to decrease the number of reference frames needed.

The main task of the mapping procedure is to provide a sparse geometrical reconstruction of the observed terrain for the UAV to navigate in it. The simultaneous calculation of the kinematics and map is described by the joint posterior:

$$p\left(\mathbf{x}_k, \mathbf{m}_k^{[1:L]} \middle| \mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{z}_k\right), \tag{2}$$

where $\mathbf{m}_k^{[1:L]}$ is a set of all L landmarks, $\mathbf{u}_k$ is a vector of IMU readings, and $\mathbf{z}_k$ is the observation vector extracted from the camera image. To decrease the number of dimensions to estimate, the above equation is commonly further transformed in accordance with the Rao-Blackwell factorization [27]. Rao-Blackwellization of the particle filter exploits dependencies between different dimensions of the state space in (2). Namely, by assuming that knowledge of consecutive robot poses is sufficient to build an individual map of landmarks, the particle filter sample set can be responsible only for representing different UAV trajectory hypotheses. Then, each particle includes individual information about its surroundings, which results in marginalization of landmarks from the estimated state space and consequently, in factorization of the posterior:

$$p\left(\mathbf{x}_k, \mathbf{m}_k^{[1:L]} \middle| \mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{z}_k\right) = p\left(\mathbf{x}_k \middle| \mathbf{x}_{k-1}, \mathbf{u}_k\right) \times \prod_{l=1}^{L} p\left(\mathbf{m}_k^l \middle| \mathbf{x}_k, \mathbf{z}_k\right). \tag{3}$$

The map constructed by the SLAM algorithm is a sparse structure consisting of landmarks ${}^{p}\mathbf{m}_k^l$ estimated by single EKFs for every tracked scene point for a given particle p. The landmark vector state is expressed using the inverse-depth point notation [28]:

$$ {}^{p}\mathbf{m}_k^l = \left[x_0 \;\; y_0 \;\; z_0 \;\; \varepsilon \;\; \alpha \;\; \rho\right]^T, \tag{4} $$

where x_0, y_0, z_0 are the coordinates of an anchor point – from which the landmark was observed for the first time. Further, ε and α are the elevation and azimuth angles, expressed in the East-North-Up (XYZ) frame, at which the scene point was registered by the camera, while $\rho = \dfrac{1}{d}$ is the inverse of the distance between the sensor and the landmark. This representation is known as *inverse-distance point* (IDP) or *anchored modified-polar points* (AMPP). An exemplary parametrization is shown in Fig. 1.

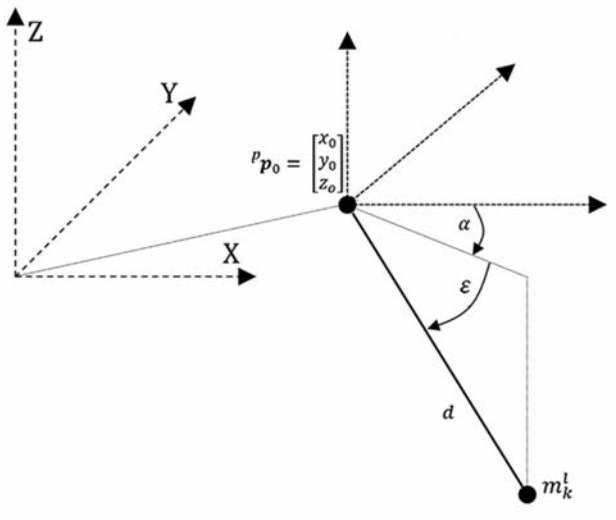

Fig. 1. IDP landmark parametrization.

Landmarks are extracted from image frames registered by the camera. To calculate 3D representation of 2D features detected in the image plane, at least two consecutive observations – from different points of view – are needed. Further, spatial relationships between the image plane and sensor's surroundings are identified using a calibrated camera whose intrinsic parameters are described by the matrix $\mathbf{K}_{\text{intr}}$:

$$ \mathbf{K}_{\text{intr}} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5} $$

where f_x and f_y represent focal lengths along the camera axes, u_0 and v_0 are the principal point offset and s is a skew of the camera axes.

There are numerous available algorithms that allow to extract good features to track, *e.g.* [29, 30] and [31]. In our implementation we choose to use an ORB detector [32], although other methods can be used interchangeably. During feature extraction, not only are the pixel coordinates of the scene points calculated, but also a descriptor. The descriptors are used to match newly observed features with already initialized landmarks constituting the map. The matching criteria

Metrol. Meas. Syst., Vol. 30 (2023), No. 3, pp. 423–439
DOI: 10.24425/mms.2023.146420

implemented in our approach seek the descriptor whose Hamming distance is minimum with respect to all other descriptors associated with the landmark expected to appear in a given camera frame. More detailed information concerning landmark initialization and update procedures can be found in [24].

Different particles represent alternative hypotheses describing both the trajectory of the UAV and the map created during its flight. To estimate the UAV state and its surroundings, one has to identify more and less probable of those hypotheses. The procedure of calculating the likelihood of different particles is called particle weighting. Samples' weights are determined by the accuracy of the predicted landmarks' location. It is calculated with respect to the current sensor pose and compared with coordinates of features extracted from the most recent camera frame – which were matched with the scene points already initialized in the map. The weight w_k^p of a given particle p in a time step k is inversely proportional to the measurement residual $\mathbf{y}_k^{[1:L_k^p]}$ of the landmarks' locations projection onto the image frame. The mathematical formula is given below:

$$w_k^p = w_{k-1}^p \prod_{l=1}^{L_k^p} \left| 2\pi \mathbf{S}_k^{[l]} \right|^{-1/2} \times \exp\left[-\frac{1}{2} \left(\mathbf{y}_k^l \right)^T \left(\mathbf{S}_k^l \right)^{-1} \mathbf{y}_k^l \right], \tag{6}$$

where w_{k-1}^p is the previous particle weight (indicating the influence of the earlier hypothesis probability evaluation), $\mathbf{S}_k^{[1:L_k^p]}$ is the innovation covariance matrix constructed for all the landmarks that were matched with the previously seen scene points and L_k^p is the number of landmarks matched by particle p.

Feature matching using only feature descriptor comparison can relatively frequently lead to mismatches and spatial outliers. This issue is solved using spatial gates which are formed in accordance with (7):

$$\sqrt{\left(\mathbf{y}_k^l \right)^T \left(\mathbf{S}_k^l \right)^{-1} \mathbf{y}_k^l} < \text{gatingThreshold}, \tag{7}$$

where the gating threshold is a measured standard deviation and is set to 3 by default.

Lastly, as different trajectory hypotheses become more diverse, the differences in weights become more significant, allowing to point out the least probable estimates. To prevent the filter from becoming increasingly less efficient, the resampling is implemented if the efficient number of particles N_{eff}, becomes smaller than the predefined threshold:

$$N_{\text{eff}} = \frac{1}{\sum_{p=1}^{N} (w^p)^2} < \text{efficientParticlesThresh}. \tag{8}$$

2.2. Homography augmentation

The extension we propose to implement in the described above SLAM algorithm is based on the idea of refining particle distribution using additional information derived from the analysis of sensor readings. Not only does the proposal distribution rely on the motion model, but it takes two recent camera measurements into consideration as well. As a result, a more efficient proposal distribution can be obtained. This leads to more robust and accurate SLAM realization. The tool we use to achieve the distribution augmentation is the homography matrix.

The homography is a relation defined between two images of the same planar surface in space. Let's assume that the images were taken from two different camera poses in space: $c1$ and

$c2$. If positions of points on the surface are identified using vectors containing the homogeneous coordinates $\begin{bmatrix} x & y & 1 \end{bmatrix}^T$ in Euclidean space, the spatial relationship (up to a scale) between them can be expressed using the Euclidean homography matrix ${}^{c2}\mathbf{H}_{c1}$:

$$\gamma \begin{bmatrix} {}^{c2}x \\ {}^{c2}y \\ 1 \end{bmatrix} = {}^{c2}\mathbf{H}_{c1} \begin{bmatrix} {}^{c1}x \\ {}^{c1}y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} {}^{c1}x \\ {}^{c1}y \\ 1 \end{bmatrix}, \tag{9}$$

where γ is the scale factor.

Fig. 2. Two views of the same surface from different camera poses.

If points are expressed in the image coordinates, using pixels $p = \begin{bmatrix} u & v & 1 \end{bmatrix}$, the Euclidean homography matrix ${}^{c2}\mathbf{H}_{c1}$ has to be replaced with a projective homography matrix ${}^{c2}\mathbf{G}_{c1}$, for the relationship to hold:

$$\gamma \begin{bmatrix} {}^{c2}u \\ {}^{c2}v \\ 1 \end{bmatrix} = {}^{c2}\mathbf{G}_{c1} \begin{bmatrix} {}^{c1}u \\ {}^{c1}v \\ 1 \end{bmatrix}. \tag{10}$$

The relationship between the matrices ${}^{c2}\mathbf{H}_{c1}$ and ${}^{c2}\mathbf{G}_{c1}$ is given in the equation below:

$$\gamma\, {}^{c2}\mathbf{G}_{c1} = \gamma \mathbf{K}_{\text{intr}}\, {}^{c2}\mathbf{H}_{c1}\, \mathbf{K}_{\text{intr}}^{-1}. \tag{11}$$

As (11) is valid up to a scale factor, ${}^{c2}\mathbf{H}_{c1}$ has only eight degrees of freedom. Thus, it is required that we have four corresponding coplanar scene points matched. At least three of those features have to be non-collinear points, since each pair of corresponding points provides two independent constraints.

Metrol. Meas. Syst., Vol. 30 (2023), No. 3, pp. 423–439
DOI: 10.24425/mms.2023.146420

The homography matrix contains information about the spatial relationship between the camera poses from which the images were taken. The relationship is encoded using a tuple consisting of:

- vector ${}^{c2}\mathbf{t}_{c1}$ – the translation between the reference frames tied to camera poses,
- matrix ${}^{c2}\mathbf{R}_{c1}$ – the rotation between the reference frames tied to camera poses,
- vector ${}^{c1}\mathbf{n}$ – the normal to the observed surface expressed in the reference frame tied to the first camera pose.

In the proposed SLAM algorithm extension, ${}^{c2}\mathbf{t}_{c1}$ and ${}^{c2}\mathbf{R}_{c1}$ are used as inputs to the correction step that refines the proposal distribution.

The procedure of calculating a tuple of vector ${}^{c2}\mathbf{t}_{c1}$, matrix ${}^{c2}\mathbf{R}_{c1}$ and vector ${}^{c1}\mathbf{n}$ from a homography matrix is called the homography matrix decomposition. There are different algorithms that perform homography decomposition, both numerical, which use *singular value decomposition* (SVD) [33] and analytical. Regardless of the chosen method, there are no less than four potential solutions to the homography decomposition. Naturally, only one tuple provides the valid transformation. In order to correctly reject invalid solutions, an additional step of eliminating of impossible solutions has to be performed. A detailed instruction on how to perform this analytical elimination can be found in [34]. In our approach, the rejection of invalid solutions is performed using the kinematics.

Commonly, in order to perform a Simultaneous Localization and Mapping procedure robustly and accurately using a particle filter, the proposal distribution of particles should match the desired distribution as closely as possible. Therefore, having samples, representing poses of the camera, sampled at the highest possible frequency – for example equal to inertial measurements data sampling rate – can be considered not sufficiently efficient, as long as the *probability mass function* (PMF) could be further refined using additional available information. We propose to resolve the potential estimation inefficiency, resulting from IMU relative inaccuracy and drift, by implementing the maximum a posteriori particle states correction using the data extracted from the homography matrix decomposition. This is performed in accordance with (12):

$$p\left(\mathbf{x}_k \middle| \mathbf{x}_{k-1},\, \mathbf{u}_k,\, \mathbf{z}_{k:k-1}\right), \tag{12}$$

where $\mathbf{z}_{k:k-1}$ is information extracted from a homography matrix describing relations of two most recent camera frames. As mentioned before, the homography only describes the relationship between different views of a given surface and does not allow to grasp the more complex relationships between points which are not coplanar. However, it was assumed, that for the adopted model of usage, where a camera-carrying UAV floats high over terrain with its gyro-stabilized sensor facing downwards, using homography would be a reasonable simplification.

The first step of our algorithm is to compare the direction of the calculated vector ${}^{c2}\mathbf{t}_{c1}$ with the direction of velocity vectors of individual particles state vectors, to determine if the result of homography matrix decomposition is valid in terms of the current PF motion estimate. The comparison is performed particle-wise, thus not every sample of an entire set is going to be corrected during a given procedure. The result of such scheme is beneficial, as one of the particle filter main advantages is its ability to simultaneously estimate diverse SLAM hypotheses – including hypotheses interpreting the validity of homography matrix decomposition differently. There are two criteria of the comparison. First, the difference in the directions of vectors, expressed as an angle, cannot be larger than a predefined upper threshold – this criterion is rather intuitive. Next, if the difference between the vectors' directions is smaller than a predefined lower threshold, the correction for a given particle is aborted as well. This mechanism is introduced to address the finite accuracy of the homography matrix calculation and its decomposition. Introduc-

ing the particle-wise comparison criteria also preserves mutual independence between different hypotheses represented by individual particles.

If the homography matrix decomposition solution is defined as valid for a given particle, the correction step is realized according to the following set of equations. First, the measurement vector $\mathbf{z}_k$ has to be defined:

$$\mathbf{z}_k = \begin{bmatrix} {}^{c2}\mathbf{t}_{c1} \\ \Delta\psi \end{bmatrix}, \tag{13}$$

where $\Delta\psi$ is the yaw angle difference between the current orientation of the UAV and the previous orientation at which the reference frame was taken. $\Delta\psi$ is extracted from ${}^{c2}\mathbf{R}_{c1}$ matrix.

Next, for each particle meeting the criteria for correction, a difference $\mathbf{\Delta}_k^p$ between the measurement and the predicted measurement $h\left(\mathbf{x}_{k|k-1}^p\right)$ is calculated:

$$\mathbf{\Delta}_k^p = \mathbf{z}_k - h\left(\mathbf{x}_{k|k-1}^p\right), \tag{14}$$

where h is the observation function, which uses a particle's state as an argument and returns a change in pose and angle between two most recently captured frames. The mathematical formula for the h is given below:

$$h\left(\mathbf{x}_{k|k-1}^p\right) = \begin{bmatrix} {}^{c2}\mathbf{R}_{\mathrm{ENU}} \begin{bmatrix} \dfrac{x_{k|k-1} - x_{k-1}}{\sqrt{\left(x_{k|k-1} - x_{k-1}\right)^2 + \left(y_{k|k-1} - y_{k-1}\right)^2 + \left(z_{k|k-1} - z_{k-1}\right)^2}} \\ \dfrac{y_{k|k-1} - y_{k-1}}{\sqrt{\left(x_{k|k-1} - x_{k-1}\right)^2 + \left(y_{k|k-1} - y_{k-1}\right)^2 + \left(z_{k|k-1} - z_{k-1}\right)^2}} \\ \dfrac{z_{k|k-1} - z_{k-1}}{\sqrt{\left(x_{k|k-1} - x_{k-1}\right)^2 + \left(y_{k|k-1} - y_{k-1}\right)^2 + \left(z_{k|k-1} - z_{k-1}\right)^2}} \\ \psi_{k|k-1} - \psi_{k-1} \end{bmatrix} \end{bmatrix}_{4\times 1}, \tag{15}$$

where the temporary translation in the local camera reference frame is calculated together with the change in the yaw angle, while k and $k-1$ address specifically the time steps during which two consecutive images were taken. ${}^{c2}\mathbf{R}_{\mathrm{ENU}}$ defines the rotation from the ENU coordinate frame to the current camera frame. We choose to drop the pitch and roll from the residual calculation as it would be inefficient to track negligible momentary rotations in horizontal axes with a gyro-stabilized camera.

In the next step of the correction procedure, it is required to calculate the Jacobi matrix for the function h:

$$\mathbf{J}_k^p = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}_{k|k-1}^p}. \tag{16}$$

$\mathbf{J}_k^p$ is a 4×9 matrix, but we choose to omit the inclusion of equation (16) closed-form solution in the paper, as it would be difficult due to the size of the Jacobian. Furthermore, it is relatively easy to calculate.

To proceed with *minimum-mean-square-error* (MMSE) state correction, one must calculate the covariance of the difference $\mathbf{\Delta}_k^p$:

$$\mathbf{S}_k^p = \mathbf{J}_k^p \, \mathbf{P}_k \, \mathbf{J}_k^{pT} + \mathbf{R}_k \, . \tag{17}$$

Metrol. Meas. Syst., Vol. 30 (2023), No. 3, pp. 423–439
DOI: 10.24425/mms.2023.146420

The covariance matrices denoted as $\mathbf{P}_k$ and $\mathbf{R}_k$ are predefined rather than calculated and allow to adjust the correction magnitude elementwise.

In the next step, the algorithm proceeds with the *maximum-a-posteriori* MAP correction gain matrix calculation:

$$\mathbf{K}_k^p = \mathbf{P}_k \, \mathbf{J}_k^{pT} \left(\mathbf{S}_k^p \right)^{-1} . \tag{18}$$

Finally, the correction of the state vector of a given particle is performed in accordance with the equation below:

$$\mathbf{x}_{k|k}^p = \mathbf{x}_{k|k-1}^p + \mathbf{K}_k^p \, \boldsymbol{\Delta}_k^p . \tag{19}$$

The procedure described by the equation above is directly derived from the Extended Kalman Filter routine. In Fig. 3, an exemplary effect of correction using information extracted from the homography matrix decomposition is presented.

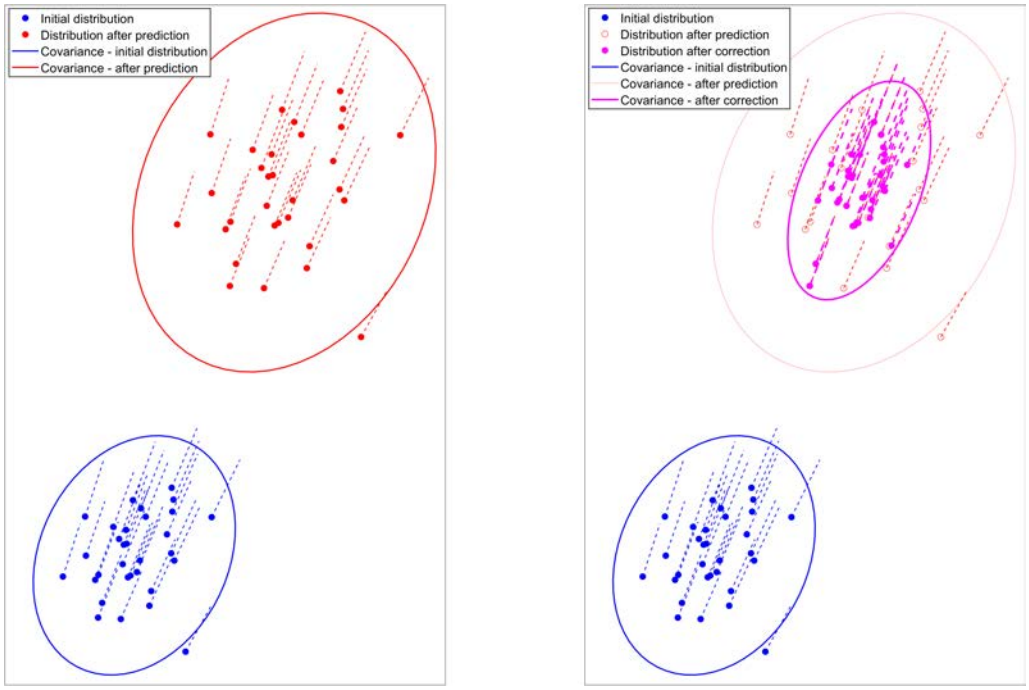

Fig. 3. Exemplary effect of the distribution refinement step. Left: Particles after prediction without the correction. Right: Particles after prediction with the correction.

2.3. Homography from displacement for outlier removal

Additionally, we introduced another homography related mechanism, which allows us to significantly mitigate the problem of ambiguous data association during feature matching. Due to changes in lighting and perspective, as well as due to the sensor noise and the surroundings appearance itself, seeking for the descriptor with the minimal Hamming distance with respect to all other descriptors is prone to generating mismatches. Including mismatches could result in erroneous homography matrix calculation. To counter this adverse effect, we choose to add an intermediate step between the feature matching and homography calculation. In this step, we

evaluate an approximate homography matrix with the displacement estimated by the particle filter and introduce an outlier removal scheme which uses the evaluated homography. This mechanism is implemented as common for every particle and uses the weighted mean state of the UAV. A kinematic transformation between two time steps during which two consecutive images were registered can be encoded using:

- rotation matrix ${}^{c2}\mathbf{R}'_{c1}$,
- translation vector ${}^{c2}\mathbf{t}'_{c1}$,

where the right superscript $'$ indicates that ${}^{c2}\mathbf{R}'_{c1}$ and ${}^{c2}\mathbf{t}'_{c1}$ were calculated using the information about a UAV displacement from the state vector $\mathbf{x}_k$. Knowing a mean depth of currently observed landmarks, we can calculate an approximate distance ${}^{c2}d$ to a surface on which the pseudo-coplanar points, extracted from the most recent image, would lay. All this information, together with a vector which is perpendicular to the observed surface – ${}^{c1}\mathbf{n}$ – can be used to estimate the homography matrix ${}^{c2}\mathbf{H}'_{c1}$ with the following formula [34]:

$$ {}^{c2}\mathbf{H}'_{c1} = {}^{c2}\mathbf{R}'_{c1} - \frac{{}^{c2}\mathbf{t}'_{c1}\,{}^{c1}\mathbf{n}^T}{{}^{c2}d} . \tag{20} $$

The relationship encoded in (20) is shown in Fig. 4.

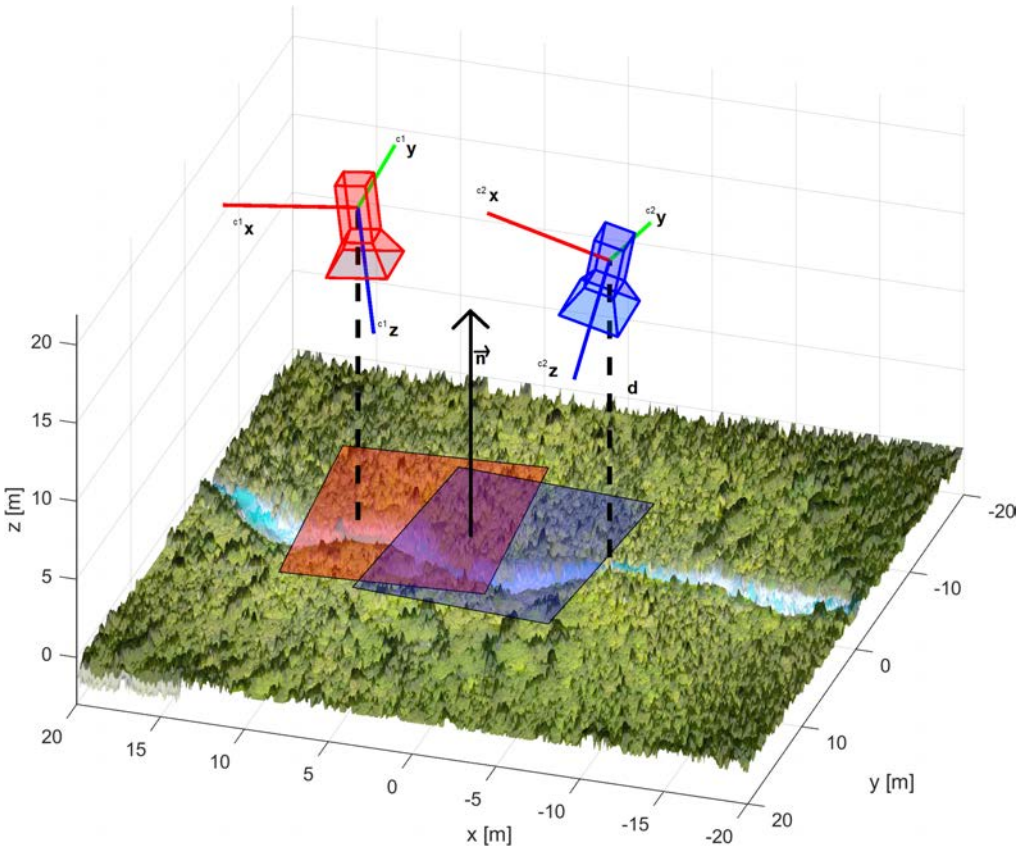

Fig. 4. Calculation of the homography matrix from the camera displacement.

Metrol. Meas. Syst., Vol. 30 (2023), No. 3, pp. 423–439
DOI: 10.24425/mms.2023.146420

Matrix ${}^{c2}\mathbf{H}'_{c1}$ is used to compare coordinates of every matched pair of points, by finding the distance between them in a common reference frame. The formula for the i-th matched pair of points is given below:

$$\text{dist}^i = \left\| \begin{bmatrix} {}^{c2}u^i \\ {}^{c2}v^i \\ 1 \end{bmatrix} - \mathbf{K}_{\text{intr}}\, {}^{c2}\mathbf{H}'_{c1}\, \mathbf{K}_{\text{intr}}^{-1} \begin{bmatrix} {}^{c2}u^i \\ {}^{c1}v^i \\ 1 \end{bmatrix} \right\|. \tag{21}$$

Next, the mean distance between points is calculated and the pairs that exceed a threshold defined in terms of the mean distance are removed.

In Figs. 5 and 6, an example of this outlier removal mechanism is presented.

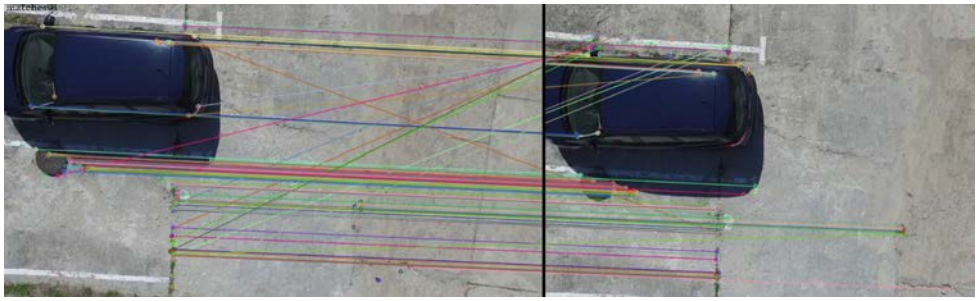

Fig. 5. Descriptors matching without outlier removal.

Fig. 6. Descriptors matching with outlier removal.

3. Experiments and results

To evaluate consequences of implementation of the proposed extension of the monocular particle filter SLAM, we conducted a series of experiments, using both simulation and real-world data. Material collected for both cases was analysed offline. The main goal of the undertaken experiments was to perform a direct comparison of the accuracy and robustness of the following SLAM approaches:

- the standalone SLAM algorithm from our previous work [24],
- the same algorithm, but with additional homography augmented proposal distribution extension.

3.1. Simulation results

For evaluating the performance of the compared methods, the Gazebo open-source software [35] was selected as the robotics simulator to generate data. The experimental setup involved two sensors following a predetermined trajectory. The camera was positioned to capture a downward view, while the IMU provided 3D accelerations and angular velocities. To ensure similarity between the real-world and simulation environment, an aerial photograph taken from a UAV was applied as a texture, effectively covering the ground plane in Gazebo.

The purpose of comparison using simulation data was to evaluate both approaches in the environment where the conditions are accurately defined and known. This provided the capability for a precise quantitative comparison of the estimated UAV trajectory with the reference, *i.e.* ground truth trajectory. Hence, the calculation of exact position and altitude errors was feasible.

To evaluate the influence of proposal distribution refinement on simulated data, we compared the performance of both approaches in setups with different numbers of particles – a relatively difficult configuration with 10 samples and a less demanding filter setting with 30 samples performing estimation.

The simulation scenario assumed that the vehicle was travelling at a constant altitude for about 75 seconds. An exemplary result of a SLAM procedure performed using simulated data trajectory is visible in Fig. 7. There are four possible loop closures along the UAV route.

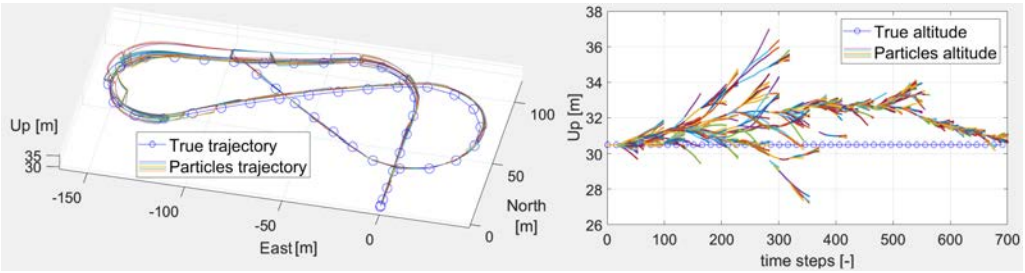

Fig. 7. Simulated SLAM routine for 30 particles – proposed method. Left: 3D trajectory – estimated and ground truth. Right: Altitude – estimated and ground truth.

Next, we performed a set of 20 runs for each configuration, where configuration is defined by the use of homography augmentation and the number of particles. For every set, we averaged the results.

To compare the algorithm accuracy, we calculated the *root mean square error* (RMSE) of the 3D trajectory estimation as well as of the estimation of the altitude exclusively. Further, we performed calculations to determine the RMSE associated with the estimation of the azimuth angle estimation.

The robustness of the approach was evaluated by analysing the number of correctly closed loops along the trajectory. The proportion of valid solutions to the homography matrix decomposition was also calculated in terms of its compliance with the estimated robot motion. The results gathered during the experiment are presented in Table 1.

As shown in Table 1, the RMSE of the trajectory estimation without the homography augmentation was 24.7 m and 14.5 m for the 10 and 30 particle configuration respectively. For the runs which included the homography augmentation, the RMSE was 10.6 m and 6.8 m respectively. The yaw angle RMSE for homography augmented SLAM was comparatively lower as well – 0.91° versus 1.05° without augmentation for 10 particles and 0.87° versus 0.96° for 30 particles.

Metrol. Meas. Syst., Vol. 30 (2023), No. 3, pp. 423–439
DOI: 10.24425/mms.2023.146420

Table 1. Comparison of simulation results.

Number of particles		10		30	
Homography augmentation		No	Yes	No	Yes
Percent of correct loop closures [%]	Loop 4	20	80	45	85
	Loop 3	55	95	75	100
	Loop 2	75	100	90	100
	Loop 1	90	100	90	100
Avg. ratio of valid solution of ${}^{c2}\mathbf{H}_{c1}$ decomposition [%]		–	86.5	–	82.7
Avg. RMSE – altitude [m]		5.7	4.6	4.7	2.7
Avg. RMSE – trajectory [m]		24.7	10.6	14.5	6.8
Avg. RMSE – yaw angle [°]		1.05	0.91	0.96	0.87

The proposed modification had a more significant influence in demanding configuration, with the lower number of particles. This indicates the superior accuracy of the proposed approach.

The number of successful loop closures was higher as well, showing a significant robustness advantage of proposal refinement using the homography matrix.

3.2. Real-world data results

In order to thoroughly evaluate the effectiveness of our approach, we conducted a comparative analysis of SLAM approaches using real-world data obtained from a UAV. The data collection process involved utilizing a DJI Matrice M100, with a Raspberry Pi serving as the onboard computer. Images were captured using a Zenmuse X3 camera, while the onboard flight controller provided the necessary kinematic data.

The main objective of the real-world data evaluation was to compare both approaches in a more demanding environment to determine whether the proposed improvement is valid outside the simulated surroundings. We aimed at confronting the approach with outdoor conditions.

The difficulties of the real-world dataset, in terms of the SLAM procedure, resulted from a number of conditions. First, it was caused by synchronization inaccuracies and sensor noise. Further, our flight scenario assumed a relatively low altitude (about 10 meters above the ground) and a flight speed up to 20 km/h, causing a short time of landmark visibility – even though the camera field of view was 94 degrees. These mentioned dataset properties, together with a blur and a sudden motion could contribute to tracking failure or map corruption.

An exemplary result of a SLAM procedure performed using real-world data trajectory is visible in Fig. 8. There are two possible loop closures along the UAV route.

We compare the same parameters except for the overall trajectory estimation error. It was omitted since the reference trajectory of a UAV was determined by an on-board autopilot unit and the calculation would lack sufficient accuracy. However, we decided to evaluate altitude RMSE. As the aircraft was piloted at a constant height, the altitude error can be a useful indicator of the altitude evaluation drift.

During the evaluation with real-world data, we compared the two SLAM approaches with three different settings concerning the number of particles – the most demanding configuration with only 10 samples and less difficult settings with 30 and 50 samples. Again, we performed a set of 20 runs for each approach – for a given number of particles – after which we averaged the results.

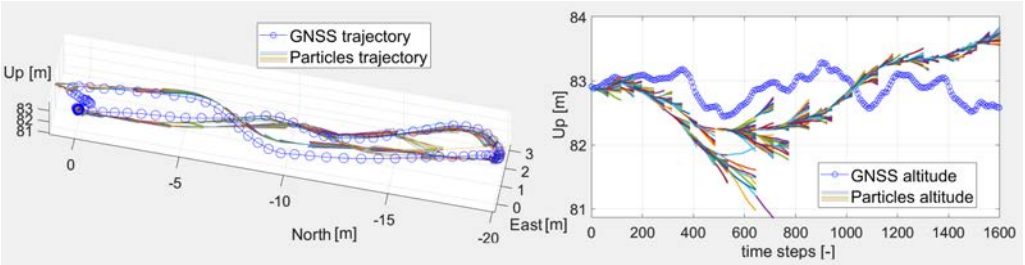

Fig. 8. Real-world SLAM routine for 30 particles – proposed method. Left: 3D trajectory – estimated and GNSS. Right: Altitude – estimated and GNSS.

The results gathered in the experiment are in Table 2.

Table 2. Comparison of real-world results.

Number of particles		10		30		50	
Homography augmentation		No	Yes	No	Yes	No	Yes
Percent of correct loop closures [%]	Loop 2	35	70	55	75	60	85
	Loop 1	55	95	70	95	80	100
Avg. ratio of valid solution of ${}^{c2}\mathbf{H}_{c1}$ decomposition [%]		–	52.1	–	55.4	–	55.8
Average RMSE – altitude [m]		7.6	4.9	6.7	4.2	4.6	2.4

As shown in Table 2, for every filter configuration with homography augmentation the percentage of successfully closed loops was higher. The RMSE of the altitude estimation without the homography augmentation was significantly larger. This indicates lower altitude drift for the proposed approach. For the algorithm without the homography augmentation, estimation errors led to tracking failure together with map corruption, which resulted in inability to perform loop closures. This again shows the increased robustness of the proposed algorithm extension.

4. Conclusions

The goal of the implemented proposal distribution refinement was to achieve a more accurate and robust SLAM algorithm. From the presented results, it can be clearly seen that the introduction of the homography matrix augmentation proved to be useful in terms of the algorithm overall performance. The particle filter utilizing information from the homography matrix decomposition focuses the particles around the correct trajectory much better.

Additional benefits can be pointed out when considering the altitude drift. In downward facing camera scenarios, at relatively high altitudes, vertical movement introduces the smallest contribution to the changes in landmark positions predicted in the pixel coordinates. In addition, upward drift causes the increase of the measurement gates size and cannot be fought using stratification [24] as efficiently as lateral movements. Hence, the positive influence of the homography refinement containing the altitude drift should be further considered advantageous.

Consideration should be given to a significantly lower success rate of homography matrix decompositions when evaluating a real-world scenario. This was potentially caused by abrupt

Metrol. Meas. Syst., Vol. 30 (2023), No. 3, pp. 423–439
DOI: 10.24425/mms.2023.146420

manoeuvres and image blurs, together with a rather strict validity threshold. However, on average, the valid solution was found every second frame, even under those difficult conditions.

Our next step is to test the presented approach with an onboard embedded system mounted on a UAV. Furthermore, since the particle filter framework enables easy integration of more sensors into a system by adding additional weighting and resampling procedures, we intend to implement our approach in a multisensory and multiplatform system with UWBs, lidars and a team of UAVs.

Acknowledgements

This work was supported by the Military University of Technology, Poland, under research project UGB 22-866.

References

[1] Durrant-Whyte, H., & Bailey, T. A. (2006). Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine*, *13*(2), 99–110. https://doi.org/10.1109/mra.2006.1638022

[2] Walter, M. J., Eustice, R. M., & Leonard, J. P. (2007). Exactly Sparse Extended Information Filters for Feature-based SLAM. *The International Journal of Robotics Research*, *26*(4), 335–359. https://doi.org/10.1177/0278364906075026

[3] Davison, A. R., Reid, I., Molton, N., & Stasse, O. (2007). MonoSLAM: Real-Time Single Camera SLAM. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, *29*(6), 1052–1067. https://doi.org/10.1109/tpami.2007.1049

[4] Cheng, J., Kim, J., Jiang, Z., & Yang, X. (2014). Compressed Unscented Kalman filter-based SLAM. *Robotics and Biomimetics*. https://doi.org/10.1109/robio.2014.7090563

[5] Forster, C., Pizzoli, M., & Scaramuzza, D. (2014). SVO: Fast semi-direct monocular visual odometry. *International Conference on Robotics and Automation*. https://doi.org/10.1109/icra.2014.6906584

[6] Engel, J., Stückler, J., & Cremers, D. (2015). Large-scale direct SLAM with stereo cameras. *Intelligent Robots and Systems*. https://doi.org/10.1109/iros.2015.7353631

[7] Mur-Artal, R., & Tardós, J. D. (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, *33*(5), 1255–1262. https://doi.org/10.1109/tro.2017.2705103

[8] Grisetti, G., Kümmerle, R., Stachniss, C., & Burgard, W. (2010). A Tutorial on Graph-Based SLAM. *IEEE Intelligent Transportation Systems Magazine*, *2*(4), 31–43. https://doi.org/10.1109/mits.2010.939925

[9] Strasdat, H., Montiel, J. M. M., & Davison, A. J. (2010). Real-time monocular SLAM: Why filter? *International Conference on Robotics and Automation*. https://doi.org/10.1109/robot.2010.5509636

[10] Montemerlo, M., Thrun, S., Roller, D., & Wegbreit, B. (2003). FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *International Joint Conference on Artificial Intelligence*, 1151–1156. https://ijcai.org/Proceedings/03/Papers/165.pdf

[11] Williams, B. W., Klein, G., & Reid, I. (2007). Real-Time SLAM Relocalisation. *International Conference on Computer Vision*. https://doi.org/10.1109/iccv.2007.4409115

[12] Avots, D., Lim, E., Thibaux, R., & Thrun, S. (2002). A probabilistic technique for simultaneous localization and door state estimation with mobile robots in dynamic environments. *Intelligent Robots and Systems*. https://doi.org/10.1109/irds.2002.1041443

[13] Zafari, F., Gkelias, A., & Leung, K. K. (2019). A Survey of Indoor Localization Systems and Technologies. *IEEE Communications Surveys and Tutorials*, *21*(3), 2568–2599. https://doi.org/10.1109/comst.2019.2911558

[14] Tateno, K., Tombari, F., Laina, I., & Navab, N. (2017). CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction. *ArXiv (Cornell University)*. https://doi.org/10.1109/cvpr.2017.695

[15] Mur-Artal, R., Montiel, J. M. M., & Tardós, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, *31*(5), 1147–1163. https://doi.org/10.1109/tro.2015.2463671

[16] Yuan, D., Qin, Y., Shen, X., & Wu, Z. (2021). A feedback weighted fusion algorithm with dynamic sensor bias correction for gyroscope array. *Metrology and Measurement Systems*, *28*(1), 161–179. https://doi.org/10.24425/mms.2021.136000

[17] Alhassan, H. M., & Ghahremani, N. A. (2021). A new predictive filter for nonlinear alignment model of stationary MEMS inertial sensors. *Metrology and Measurement Systems*, *28*(4), 673–691. https://doi.org/10.24425/mms.2021.137702

[18] Stawowy, M., Duer, S., Paś, J. & Wawrzyński, W. (2021). Determining Information Quality in ICT Systems. *Energies*, *14*(17). 1–18. https://doi.org/10.3390/en14175549

[19] Von Stumberg, L., Usenko, V. C., & Cremers, D. (2018). Direct Sparse Visual-Inertial Odometry Using Dynamic Marginalization. *ArXiv (Cornell University)*. https://doi.org/10.1109/icra.2018.8462905

[20] Unser, M., Omari, S., Hutter, M., & Siegwart, R. (2015). Robust visual inertial odometry using a direct EKF-based approach. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. https://doi.org/10.1109/iros.2015.7353389

[21] Yin, H., Li, S., Tao, Y., Guo, J., & Huang, B. (2022). Dynam-SLAM: An Accurate, Robust Stereo Visual-Inertial SLAM Method in Dynamic Environments. *IEEE Transactions on Robotics*, *39*(1), 289–308. https://doi.org/10.1109/tro.2022.3199087

[22] Kerl, C., Sturm, J., & Cremers, D. (2013). Dense visual SLAM for RGB-D cameras. *Intelligent Robots and Systems*. https://doi.org/10.1109/iros.2013.6696650

[23] Schops, T., Sattler, T., & Pollefeys, M. (2019). BAD SLAM: Bundle Adjusted Direct RGB-D SLAM. *Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2019.00022

[24] Słowak, P., & Kaniewski, P. (2021). Stratified Particle Filter Monocular SLAM. *Remote Sensing*, *13*(16), 3233. https://doi.org/10.3390/rs13163233

[25] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J. L., Reid, I. R., & Leonard, J. P. (2016). Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332. https://doi.org/10.1109/tro.2016.2624754

[26] Barros, A. M., Michel, M., Moline, Y., Corre, G., & Carrel, F. (2022). A Comprehensive Survey of Visual SLAM Algorithms. *Robotics*, *11*(1), 24. https://doi.org/10.3390/robotics11010024

[27] Murphy, K. (1999). Bayesian Map Learning in Dynamic Environments. *Neural Information Processing Systems*, *12*, 1015–1021. https://papers.nips.cc/paper/1716-bayesian-map-learning-in-dynamic-environments.pdf

[28] Montiel, J. M. M., Civera, J., & Davison, A. J. (2006). Unified Inverse Depth Parametrization for Monocular SLAM. *Robotics: Science and Systems*. https://doi.org/10.15607/rss.2006.ii.011

Metrol. Meas. Syst., Vol. 30 (2023), No. 3, pp. 423–439
DOI: 10.24425/mms.2023.146420

[29] Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded Up Robust Features. *Lecture Notes in Computer Science*, 404–417. https://doi.org/10.1007/11744023_32

[30] Leutenegger, S., Chli, M., & Siegwart, R. (2011). BRISK: Binary Robust invariant scalable keypoints. *International Conference on Computer Vision*. https://doi.org/10.1109/iccv.2011.6126542

[31] Lowe, D. J. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. https://doi.org/10.1023/b:visi.0000029664.99615.94

[32] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *International Conference on Computer Vision*. https://doi.org/10.1109/iccv.2011.6126544

[33] Page, G. (2005). Multiple View Geometry in Computer Vision, by Richard Hartley and Andrew Zisserman, CUP, Cambridge. *Robotica*, *23*(2), 271. https://doi.org/10.1017/s0263574705211621

[34] Malis, E., & Vargas Villanueva, M. (2007). Deeper understanding of the homography decomposition for vision-based control. *INRIA*. https://hal.inria.fr/inria-00174036/document

[35] Koenig, N., & Howard, A. (2004, September). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ international conference on intelligent robots and systems* (IROS)(IEEE Cat. No. 04CH37566) (Vol. 3, pp. 2149–2154). IEEE. https://doi.org/10.1109/IROS.2004.1389727

Paweł Leszek Słowak received his B.Sc. and M.Sc. degrees in mechatronics from the Military University of Technology in Warsaw, Poland in 2014 and 2015 respectively. He is currently an assistant lecturer and researcher at the Institute of Radioelectronics, Faculty of Electronics at the Military University of Technology. His research interests include simultaneous localization and mapping, multi-sensor fusion, navigation as well as radar target tracking.

Piotr Kaniewski studied radiotechnical systems of aircraft at the Military University of Technology in Warsaw. He graduated and received his M.Sc. in 1994, Ph.D. in 1998, and was habilitated in 2011. He worked as an Engineer, Assistant, Assistant Professor, and currently works as an Associate Professor at the Faculty of Electronics at the Military University of Technology, where, since 2012, he has been the Director of the Institute of Radioelectronics. His current research is focused on navigation systems dedicated for special purposes, such as supporting synthetic aperture radars (MOCO), supporting ground penetrating radars (GPR) with accurate scanning trajectory information, distributed navigation algorithms for UAV swarms, and navigation systems for GNSS denied environments, especially using SLAM on UAVs and UWB ranging modules for indoor navigation. He is an author of more than 200 scientific papers and 2 books.