ISSN 0005-1179, Automation and Remote Control, 2016, Vol. 77, No. 2, pp. 297–312. © Pleiades Publishing, Ltd., 2016.
Original Russian Text © D.S. Krivokon, A.T. Vakhitov, O.N. Granichin, 2016, published in Avtomatika i Telemekhanika, 2016, No. 2, pp. 142–161.

TOPICAL ISSUE

Estimating the Position of a Moving Object Based on Test Disturbance of Camera Position

D. S. Krivokon, A. T. Vakhitov, and O. N. Granichin

St. Petersburg State University, St. Petersburg, Russia
e-mail: dmitry00@gmail.com, alexander.vakhitov@gmail.com, oleg_granichin@mail.ru

Received March 30, 2015

Abstract—The problem of estimating the coordinates of a moving object based on visual data arises in numerous applications, starting from robotic and ending with the consumer market of portable devices. Traditional algorithms for solving this problem require either additional devices or significant constraints on the possible motion of the object. In this work, we present a new approach to tracking the object that lets us estimate its position under sufficiently general conditions. The method is based on randomizing the camera location independently of the object's motion; since the test disturbance we choose is independent, it lets us construct a feasible iterative pseudogradient estimation algorithm.

DOI: 10.1134/S0005117916020065

1. INTRODUCTION

Traditional approaches to estimating the coordinates of a moving object mostly either require additional resources or impose constraints on the observation conditions. For instance, popular methods include those that use two or more synchronized cameras [1, 2]. Such approaches use simple methods for triangulating the position of three-dimensional points [3] and construct the object motion's trajectory with this triangulation. To ensure the accuracy of the resulting estimates, mutual positions of the cameras have to be constantly calibrated, which is often a hard problem in itself. As an alternative to additional cameras, surveillance systems can be constructed with specialized sensors. Such sensors may include, for instance, a camera combined with an infrared projector [4]. The projector projects onto the scene in the infrared range a predefined unique pattern of dots with a fixed frequency (usually 60 Hz); the pattern is then detected by the camera, and coordinates of reference points on the pattern are reconstructed by triangulation techniques based on known coordinates of the dots' projections on the camera and the projector. Laser range finders are also often used to evaluate position [5]; they compute distance by measuring the time when a ray of light returns from the source to the detector. A different alternative in solving the problem would be to use additional constraints on the character of the object's motion. This is usually mostly relevant to outdoor surveillance applications where one can assume, for instance, that observed objects move along a plane [6] (guarding systems) or with constant speed [7] (traffic surveillance).

In this work we develop a new approach proposed in [8, 9], where we can get an estimate of the object's position by its image from a single camera without any significant constraints on the character of its motion. The use of only one camera significantly simplifies the surveillance system, thus extending its applicability. The proposed algorithm also significantly differs from reconstruction methods with a single camera [10] that consider in the original problem setting only static objects. The new approach is based on more fundamental ideas of using randomized (search) stochastic approximation algorithms (RSAA) that were first proposed in the works of O.N. Granichin [11–13], B.T. Polyak, A.B. Tsybakov [14], and J.C. Spall [15]. These methods

solve estimation and identification problems for "almost arbitrary noise" (e.g., for *unknown but bounded* deterministic noise). Algorithms of RSAA type consist of the test disturbance generation stage (the test disturbance must be independent on the noise), the data collection stage, and the gradient approximation (computing the pseudogradient) towards which the previous estimate is then modified. For the problem of estimating the coordinates of a moving object, stochastic approximation algorithms with constant step size that solve the tracking problem for the minimum of a nonstationary functional [16–18] are most relevant. Modern studies [19–23] of general properties of estimates on randomized (search) stochastic approximation algorithms with constant step size have motivated us to apply them to the problem of estimating the trajectory of a moving object by its image from a single camera that randomly changes its location. This work is devoted to this approach.

The paper is organized as follows. The introduction is followed by a section where we present the formal observation model with a single camera; Section 3 contains the problem setting and presents the basic idea of the algorithm. In Section 4, we formulate the algorithm, in Section 5 we present the main theoretic result on the upper bound on the mean squared error of the algorithm's estimates. Experimental modeling results of this work and testing in real life conditions are presented in Section 6. Section 7 concludes the paper.

2. OBSERVATION MODEL

Usually in object coordinates estimation problems with a camera the observation model is the "pinhole model" [3] with a camera whose lens is represented as a small hole in some opaque box with a light-sensitive element inside. The camera's location is defined by its optic center vector $C = (C^{(1)}, C^{(2)}, C^{(3)})^{\mathrm{T}}$ that represents the location of this hole, the focal distance f that defines the distance from optic center C to the camera plane, and the camera's angular position defined with the rotation matrix $R \in \mathbb{R}^{3\times 3}$. The rotation matrix and optic center coordinates define a transformation from some fixed global coordinate system to the local coordinate system of the camera. Here and in what follows $^{\mathrm{T}}$ denotes transposition, $\mathbb{R}$ is the space of real numbers, E denotes expectation. Observable variables in the model include object projections that are located in the camera's field of view.

In the simplest case object coordinates are given by some point $P = (P^{(1)}, P^{(2)}, P^{(3)})^{\mathrm{T}}$. To find the projection of point P on the camera plane we have to draw a ray from the point P to the

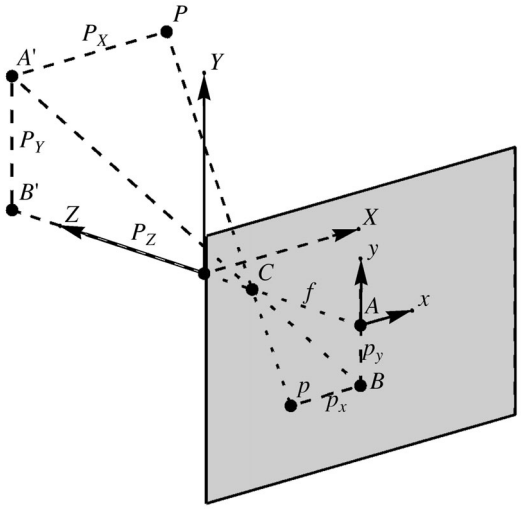

Fig. 1. The camera model.

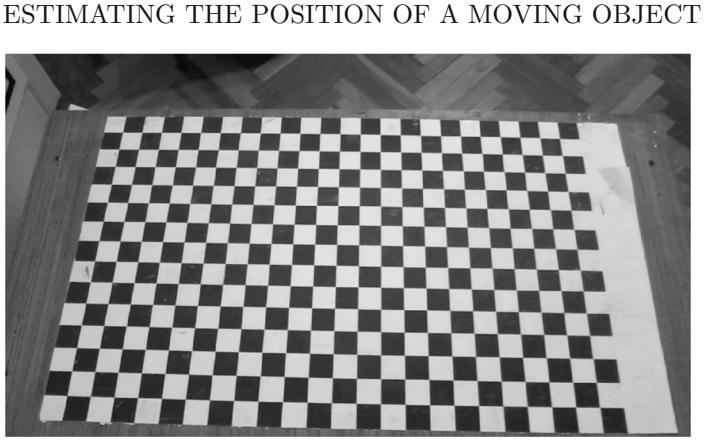

Fig. 2. A sample object used to calibrate the camera.

optic center C. The projection will be given by the point where this ray will cross the camera plane. Figure 1 schematically shows the entire process for the case when $C = (0, 0, 0)^{\mathrm{T}}$. To find coordinates of the projection onto the camera plane $p = (p^{(1)}, p^{(2)})^{\mathrm{T}}$ one can use simple properties of homothetic triangles. Considering homothetic triangles ABC and $A'B'C$, we get

$$p^{(2)} = f\frac{P^{(2)}}{P^{(3)}}.$$

We can similarly derive $p^{(1)} = f\frac{P^{(1)}}{P^{(3)}}$. If the rotation matrix is not unit, the corresponding equations look like

$$p^{(1)} = f\frac{R_{11}P^{(1)} + R_{12}P^{(2)} + R_{13}P^{(3)} + t^{(1)}}{R_{31}P^{(1)} + R_{32}P^{(2)} + R_{33}P^{(3)} + t^{(3)}},$$
$$p^{(2)} = f\frac{R_{21}P^{(1)} + R_{22}P^{(2)} + R_{23}P^{(3)} + t^{(2)}}{R_{31}P^{(1)} + R_{32}P^{(2)} + R_{33}P^{(3)} + t^{(3)}},$$

where vector $(t^{(1)}, t^{(2)}, t^{(3)})^{\mathrm{T}} = -R^{-1}(C^{(1)}, C^{(2)}, C^{(3)})^{\mathrm{T}}$.

Real life cameras seldom can fully satisfy the presented model due to their use of lenses. To correct for the optical distortions researchers use various nonlinear projection coordinate correction models. One popular such model is the model presented in [24] that does correction according to the following rule:

$$p_d^{(1)} = p_u^{(1)}(1 + K_1r^2 + K_2r^4 + \ldots) + \left(P_2(r^2 + 2{p_u^{(1)}}^2) + 2P_1p_u^{(1)}p_u^{(2)}\right)(1 + P_3r^2 + P_4r^4 + \ldots),$$
$$p_d^{(2)} = p_u^{(2)}(1 + K_1r^2 + K_2r^4 + \ldots) + \left(P_2(r^2 + 2{p_u^{(2)}}^2) + 2P_1p_u^{(1)}p_u^{(2)}\right)(1 + P_3r^2 + P_4r^4 + \ldots),$$

where $(p_u^{(1)}, p_u^{(2)})^{\mathrm{T}}$ are projection coordinates without optical distortions, and $(p_d^{(1)}, p_d^{(2)})^{\mathrm{T}}$ are distorted coordinates, $r = \sqrt{(p_u^{(1)})^2 + (p_u^{(2)})^2}$. Coefficients $K_1, K_2, \ldots, P_1, P_2, \ldots$ are called *distortion* coefficients, their number regulates the accuracy of the resulting model. Together with the focal distance, these coefficients are called internal parameters of the camera. The location of the camera's canter and the rotation matrix are called external parameters of the camera. Internal parameters are assumed to be fixed during the entire time of the observation process, and they are usually computed in advance during the calibration process. For calibration one can use an object with a known three-dimensional structure, say a chessboard (Fig. 2). Using automatic detection techniques for points of the chessboard [25] and knowing the sizes of the cells one can fit

internal parameters by minimizing projection error $(p_d^{(1)} - \hat{p}_d^{(1)})^2 + (p_d^{(2)} - \hat{p}_d^{(2)})^2$, where $(p_d^{(1)}, p_d^{(2)})^{\mathrm{T}}$ are projection coordinates returned by the chessboard detector, and $(\hat{p}_d^{(1)}, \hat{p}_d^{(2)})^{\mathrm{T}}$ are their estimates obtained with current values of the internal parameters. This kind of approach to calibration has been described in [26]. External parameters of a camera can also be computed with objects with known scene geometry, but such an approach will hardly be practical. Therefore, commonly used methods usually estimate both external camera parameters and visible static objects at the same time. A standard scheme of such an approach is shown in [2]. In what follows we will assume that all internal and external camera parameters are known. To simplify the algorithm's formulation we will assume that the camera's rotation matrix, its focal distance, and distortion coefficients are unit. We can do it without loss of generality because one can first fix all optical distortions with the model above, divide the coordinates of observed projections by the focal distance, and obtain a projective transformation for the projection coordinates from the rotation matrix that reduces them to the case of unit rotation matrix [3, 27].

3. IDEA OF THE ALGORITHM

Let external and internal parameters of the camera be known, and let $n = 1, 2, \ldots$ be discrete time. As the observed object at time moment n we consider a point with coordinate vector $P_n = (P_n^{(1)}, P_n^{(2)}, P_n^{(3)})^{\mathrm{T}}$. Coordinates of its "noisy" projection to the camera plane $p_n = (p_n^{(1)}, p_n^{(2)})^{\mathrm{T}}$ satisfy equations

$$p_n^{(1)} = \frac{P_n^{(1)} - C_n^{(1)}}{P_n^{(3)} - C_n^{(3)}} + v_n^{(1)},$$
$$p_n^{(2)} = \frac{P_n^{(2)} - C_n^{(2)}}{P_n^{(3)} - C_n^{(3)}} + v_n^{(2)},$$

where $v_n = (v_n^{(1)}, v_n^{(2)})^{\mathrm{T}}$ is the vector of unknown noises.

It is required to construct estimates of the motion's trajectory for the point P_n based on known coordinates of its projections p_n at every time moment n.

Example. Consider the case when camera position is given as $C_n = (\Delta_n^{(1)}, 0, 0)^{\mathrm{T}}$, where $\Delta_n^{(1)}$ is a known random test disturbance along the x axis with zero expectation and variance $\sigma_1^2 = E{\Delta_n^{(1)}}^2$ independent of the position of the point P_n and noise vector v_n. Equations for projections take the form

$$p_n^{(1)} = \frac{P_n^{(1)} - \Delta_n^{(1)}}{P_n^{(3)}} + v_n^{(1)},$$
$$p_n^{(2)} = \frac{P_n^{(2)}}{P_n^{(3)}} + v_n^{(2)}.$$

Multiplying the first expression by $p_n^{(1)}$ by $\Delta_n^{(1)}$, we get

$$p_n^{(1)}\Delta_n^{(1)} = \frac{P_n^{(1)}\Delta_n^{(1)}}{P_n^{(3)}} - \frac{\Delta_n^{(1)}\Delta_n^{(1)}}{P_n^{(3)}} + v_n^{(1)}\Delta_n^{(1)}.$$

Since $\Delta_n^{(1)}$ are centered and independent of the position of the point P_n and noise $v_n^{(1)}$, for the expectation we derive step by step that

$$Ep_n^{(1)}\Delta_n^{(1)} = E\left[\frac{P_n^{(1)}\Delta_n^{(1)}}{P_n^{(3)}} - \frac{\Delta_n^{(1)}\Delta_n^{(1)}}{P_n^{(3)}} + v_n^{(1)}\Delta_n^{(1)}\right] = -\sigma_1^2 E\frac{1}{P_n^{(3)}}, \tag{1}$$

since $E\frac{P_n^{(1)}\Delta_n^{(1)}}{P_n^{(3)}} = E\frac{P_n^{(1)}}{P_n^{(3)}}E\Delta_n^{(1)} = 0$ and $Ev_n^{(1)}\Delta_n^{(1)} = Ev_n^{(1)}E\Delta_n^{(1)} = 0$.

Equations (1) show that, averaging $p_n^{(1)}\Delta_n^{(1)}$ over time, we can get a value equal to the average inverse depth of the observed point $\Gamma_n = \frac{1}{P_n^{(3)}}$ multiplied by some known constant. Knowing the inverse depth, the object coordinates can be found as

$$P_n^{(1)} = \frac{p_n^{(1)}}{\Gamma_n} + \Delta_n^{(1)}, \quad P_n^{(2)} = \frac{p_n^{(2)}}{\Gamma_n}, \quad P_n^{(3)} = \frac{1}{\Gamma_n}.$$

This example intuitively illustrates the main result of this work. The algorithm we propose below results by considering a certain dynamical system where observations are represented in a form similar to $p_n^{(1)}\Delta_n^{(1)}$. Inverse depth estimates for the observed point can be computed with a simple pseudogradient method with fixes step according to the above logic.

4. PROBLEM SETTING AND ALGORITHM FOR SOLVING THE PROBLEM

Consider a coordinate system defined with respect to the position and spatial orientation of the camera. We assume that the observed object freely drifts (moves) in the three-dimensional space. Let $n = 1, 2, \ldots$ be the discrete time and let object coordinates at time moment n with respect to the camera be equal to $P_n^{(1)}, P_n^{(2)}, P_n^{(3)}$. We denote the vector of object coordinates by $P_n = (P_n^{(1)}, P_n^{(2)}, P_n^{(3)})^{\mathrm{T}}$ and consider the dynamics of its change with time. We will assume that at every time moment $n = 1, 2, \ldots$ coordinates of vector P_n change by the values $\xi_n = (\xi_n^{(1)}, \xi_n^{(2)}, \xi_n^{(3)})^{\mathrm{T}}$. The magnitude of this change depends both on the velocity of the object itself and on the discretization interval. For instance, if we get new frames with low frequency changes in the object's position will be larger, which will complicate the estimation problem. Similar to the previous section, internal and external camera parameters are assumed to be known, and an observation is a noisy projection of the point to the camera plane.

Suppose that with the same discretization frequency the at every time moment $n = 1, 2, \ldots$ moves, and this motion is given by a measurable vector $\Delta_n = (\Delta_n^{(1)}, \Delta_n^{(2)}, \Delta_n^{(3)})^{\mathrm{T}}$. Here the sequence $\{\Delta_n\}$ represents a realization of independent identically distributed random vectors with distribution symmetric with respect to zero that are independent of the object's position and noise in the measurements of projection points on the camera plane.

In the above notation, a change in the object position over time satisfies equation

$$\begin{pmatrix} P_n^{(1)} \\ P_n^{(2)} \\ P_n^{(3)} \end{pmatrix} = \begin{pmatrix} P_{n-1}^{(1)} \\ P_{n-1}^{(2)} \\ P_{n-1}^{(3)} \end{pmatrix} + \Delta_n + \xi_n, \tag{2}$$

and the coordinates of its "noisy" projection to the camera plane $p_n = (p_n^{(1)}, p_n^{(2)})^{\mathrm{T}}$ satisfy equations

$$p_n^{(1)} = \frac{P_n^{(1)}}{P_n^{(3)}} + v_n^{(1)}, \quad p_n^{(2)} = \frac{P_n^{(2)}}{P_n^{(3)}} + v_n^{(2)}.$$

We are to find the distance from the object to the camera plane at every time moment.

We make the following assumptions for $n = 1, 2, \ldots$.

1. The observed object always resides in the camera's field of view, i.e., there exist H_1 and H_2 such that $|\frac{P_n^{(i)}}{P_n^{(3)}}| < H_i$, $i = 1, 2$. Parameters H_1 and H_2, in essence, correspond to physical size of the camera's light-sensitive element, i.e., width and height respectively.

2. The inverse value of the depth of a point $\Gamma_n = \frac{1}{P_n^{(3)}}$ lies in a finite interval: $\Gamma_n \in M_\Gamma = (\Gamma_{\min}, \Gamma_{\max})$ and $\Gamma_{\min}, \Gamma_{\max} > 0$. Bounds $\Gamma_{\min}$ and $\Gamma_{\max}$ show how far from or near to the camera the object may be located. (It will be clear from Theorem 1 below that the closer an object is, the more accurate the bounds will be, which corresponds to our intuition.)

3. The drift of a point is bounded: $|\xi_n^{(1)}| < D_1$, $|\xi_n^{(2)}| < D_2$, $|\xi_n^{(3)}| < D_3$. Coefficients D_1, D_2, D_3 are directly related to discretization time and the object's own velocity. They will also influence the upper bound of the algorithm's accuracy. (This is the only constraint on the object's motion.)

4. The sequence of random changes in camera location $\{\Delta_n\}$ is a measurable realization of random vectors in $\mathbb{R}^3$ that are mutually independent and independent of the current object and observation errors, with independent components and finite statistical moments: $E|\Delta_n^{(i)}| < \delta_i < \infty$, $E|\Delta_n^{(i)}|^2 \leqslant \sigma_i^2 < \infty$ for $i = 1, 2, 3$, $E|\Delta_n^{(1)}|^4 \leqslant \sigma_4^4 < \infty$. Besides, random values $\Delta_n^{(1)}$ have a symmetric distribution and $\sigma_1^2 > 0$.

5. Additive noise is bounded by

$$|v_n^{(1)}| < c_1, \quad |v_n^{(2)}| < c_2, \quad |v_n^{(1)} - v_{n-1}^{(1)}| < c_d,$$

or, if additive noise has random nature, it is independent of Δ_n and bounded in mean square:

$$E|v_n^{(1)}| < c_1, \quad E|v_n^{(2)}| < c_2, \quad E|v_n^{(1)} - v_{n-1}^{(1)}|^2 < c_d^2.$$

Consider the following algorithm:

- set $n := 0$, choose algorithm step size $\alpha > 0$, choose an arbitrary initial approximation for the inverse distance depth value $\hat{\Gamma}_0 \in M_\Gamma$;
- iteration $n := n + 1$;
- generate and implement a test disturbance for the camera location $\Delta_n \in \mathbb{R}^3$;
- make observations $p_n = (p_n^{(1)}, p_n^{(2)})^{\mathrm{T}}$ (read from the camera);
- compute the "pseudogradient" g_n as

$$g_n = \hat{\Gamma}_{n-1} - \frac{\Delta_n^{(1)}}{\sigma_1^2}(p_n^{(1)} - p_{n-1}^{(1)}); \tag{3}$$

- update the estimate of the inverse distance depth according to the basic stochastic approximation algorithm with constant step size

$$\hat{\Gamma}_n = Pr_{M_\Gamma}(\hat{\Gamma}_{n-1} - \alpha g_n), \tag{4}$$

where $Pr_{M_\Gamma}(\cdot)$ is the projection operation to the interval M_Γ:

$$Pr_{M_\Gamma}(x) = \begin{cases} \Gamma_{\min}, & x < \Gamma_{\min} \\ \Gamma_{\max}, & x > \Gamma_{\max} \\ x & \text{otherwise;} \end{cases}$$

- compute current estimates of object coordinates

$$\hat{P}_n^{(1)} = \frac{p_n^{(1)}}{\hat{\Gamma}_n}, \quad \hat{P}_n^{(2)} = \frac{p_n^{(2)}}{\hat{\Gamma}_n}, \quad \hat{P}_n^{(3)} = \frac{1}{\hat{\Gamma}_n}. \tag{5}$$

Note that the Eq. (3) for the pseudogradient is a development of the motivating idea from the previous section, but instead of $\Delta_n \frac{P_n^{(1)}}{P_n^{(3)}}$ we consider the product of Δ_n by the difference of two sequential observations, which lets us also obtain inverse depth estimates with smaller variance. This approach is similar to RSAA type algorithms with two observations [21, 23].

5. MAIN RESULT

The following theorem establishes the bounds for estimation errors with the proposed algorithm.

Theorem. *Suppose that assumptions 1–5 are satisfied. We denote*

$$b = (1+\alpha+\alpha^2)\Gamma_{\max}^2(D_3+\delta_3),$$

$$k = (1+\alpha^2)\Gamma_{\max}^4(D_3^2+\sigma_3^2) + 2\alpha^2\Gamma_{\max}^2 + \alpha^2\left(\frac{\sigma_2^2}{\sigma_1^4}(2H_1+2c_d)^2 + \frac{\sigma_4^4}{\sigma_1^4}\Gamma_{\max}^2\right).$$

If $0<\alpha<1$ *then*

$$\overline{\lim_{n\to\infty}}\sqrt{E\|\hat{\Gamma}_n-\Gamma_n\|^2} \leqslant L = \frac{b+\sqrt{b^2+\alpha(1-\alpha)k}}{\alpha\,(1-\alpha)} \tag{6}$$

and

$$\overline{\lim_{n\to\infty}}E|\hat{P}_n^{(i)}-P_n^{(i)}| \leqslant \Gamma_{\min}^{-2}H_iL + \Gamma_{\min}^{-1}c_i, \quad i=1,2. \tag{7}$$

Proof of theorem is given in the Appendix.

6. EXPERIMENTAL RESULTS

6.1. Experiment Settings

In our modeling, we considered the case of a camera watching a single point. The discretization period was set to one. As above, C_n denotes the camera position at time moment n. To simplify the experiment, we did not consider the case of camera rotating around its axis; instead, we assumed that the camera moves with some velocity V_n^C which is known but not controlled by the user (for instance, it could be the speed of the car where the camera is installed). Under these conditions, the camera location change dynamics is given by equation $C_n = C_{n-1} + V_n^C$.

The camera watches over an object with coordinates $P_n = (P_n^{(1)}, P_n^{(2)}, P_n^{(3)})^{\mathrm{T}}$ that moves with velocity V_n^P:

$$P_n = P_{n-1} - V_n^C + V_n^P.$$

Observation error v_n is uniform, has zero expectation, and takes values in both coordinates with equal probabilities in the range of $[-0.001; 0.001]$. Note that the admissible level of noise is important: if the observed object shifts by vector $(1;0;0)^{\mathrm{T}}$ in the camera plane it will correspond to a shift of size $(\frac{1}{10};0;0)^{\mathrm{T}}$ (judging by projection equations), and the above error may constitute up to 1% of the observed distance.

To apply the algorithm above to current camera location at every time moment, we add a test randomized disturbance Δ_n:

$$C_n = C_{n-1} + V_n^C - \Delta_n,$$

where $\{\Delta_n\}$ is a sequence of realizations of a random vector on the sphere of radius 0.1:

$$\Delta_n = 0.1 \times (\sin(\Gamma_n)\cos(\phi_n), \sin(\Gamma_n)\sin(\phi_n), \cos(\Gamma_n))^{\mathrm{T}}.$$

Here $\{\phi_n\}$ and $\{\Gamma_n\}$ are sequences of independent random variables that are uniformly distributed on intervals $[0, 2\pi]$ and $[0, \pi]$ respectively. We chose the sphere's radius to be 0.1 based on the assumptions on the scale of other objects in the system and the observation noise level to be such that the observed distance between object projections at different moments of time would be no less than the value of the possible noise. In practice, we have to start from similar considerations, taking into account physical restrictions of the device that will be implementing the camera's motion.

In all experiments, the initial camera position is set to be zero $C_0 = (0;0;0)^{\mathrm{T}}$, and the initial object position is $P_0 = (0;0;10)^{\mathrm{T}}$.

6.2. Modeling Results

In the simplest case we considered the case when $V_n^C = 0$ and $V_n^P = 0$. We tested various kinds of noise in the initial approximation. In case when the initial error is small the algorithm successfully tracks the true value of the inverse depth (see Fig. 3). If, on the other hand, the error in the initial

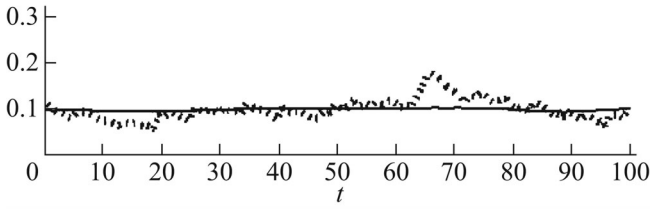

Fig. 3. Sample true trajectories (solid line) and algorithm estimates for the simple case without point movement and small error in the initial approximation.

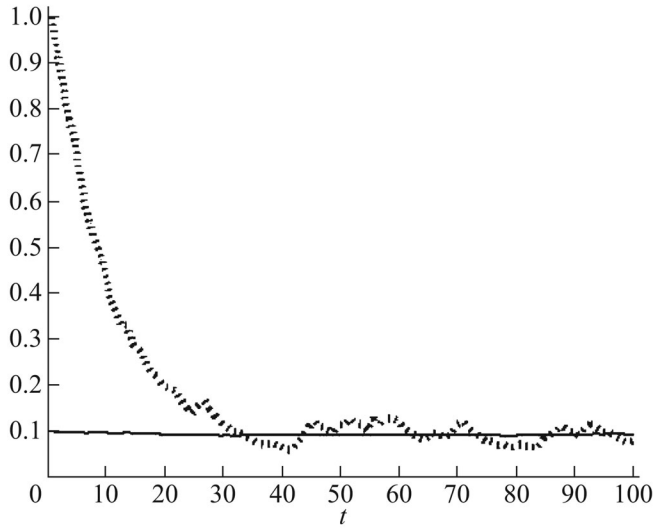

Fig. 4. Conditions similar to Fig. 3 but with large initial approximation error.

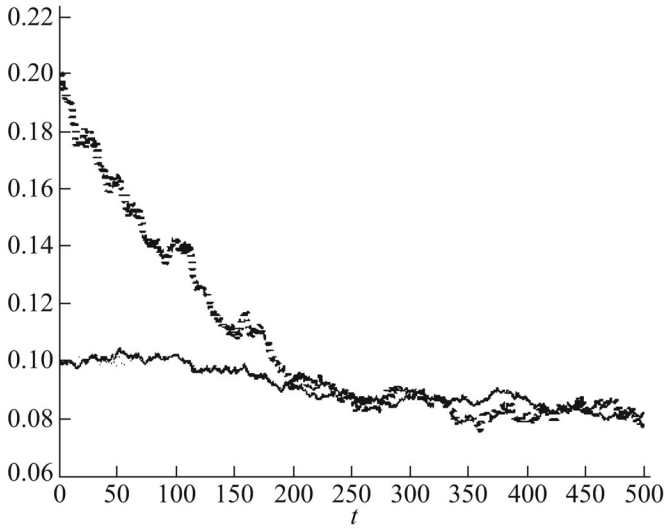

Fig. 5. Sample trajectories in case of a moving point.

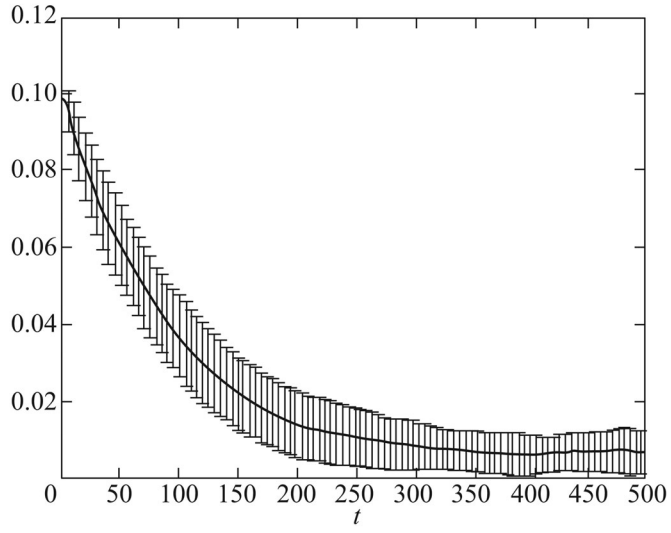

Fig. 6. Confidence intervals with deviation 3σ.

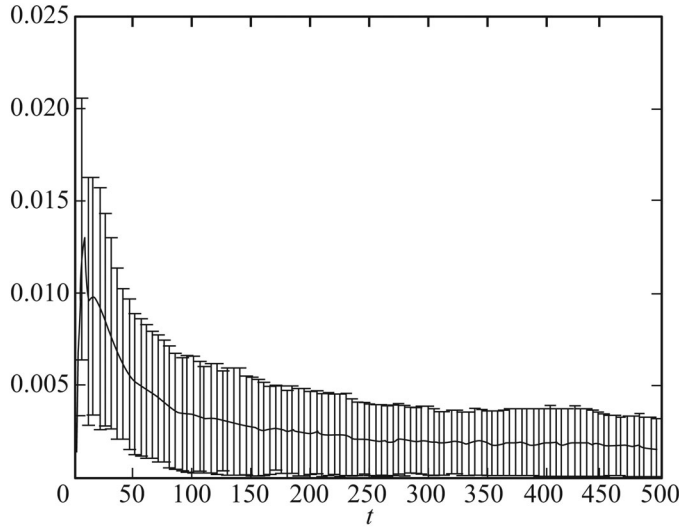

Fig. 7. Mean error for the estimate $P^{(1)}/P^{(3)}$.

approximation is large then the algorithm on first iterations converges to the true trajectory and then keeps tracking it (Fig. 4). We used the following parameters: $\hat{\Gamma}_0 = 1.1$; $\alpha = 0.1$.

The most interesting testing scenario is the case when both the point and the camera move in the same direction (Fig. 5). Under these conditions, standard computer vision techniques do not help in determining the object location. We have considered the case of $V_n^C = V_n^P = (0.5; 0; 0)^{\mathrm{T}}$. Figure 6 shows the mean error of estimating the inverse depth together with its variance computed over 1000 algorithm test runs in identical conditions ($\alpha = 0.001$). Figure 7 shows the graph of error estimates for $P^{(1)}/P^{(3)}$; the graph illustrates the fact that the algorithm also successfully handles the problem of estimating this variable. Initial approximation in the tests was chosen to be $\hat{\Gamma}_0 = 0.2$. Analyzing the graphs, we see that even in such a hard case the algorithm can produce estimates converging to the true motion trajectory of the point. We chose parameter α in different cases depending on the character of the object's motion in such a way that the resulting accuracy

would be approximately the same, and convergence would be sufficiently fast. Therefore, in case of a static object we chose a large α (to improve the convergence rate), and in case of a moving object α was small (to ensure the desired accuracy).

6.3. Real Experiment

For the purposes of illustration, we also performed an experiment on real data. Testing in real life conditions is complicated by the lack of exact information regarding the true location of the moving object and regarding external camera parameters. To get the parameters, we used a set of static markers with known geometry (Fig. 8). These markers are convenient because

Fig. 8. A base of markers used to tie camera and object location.

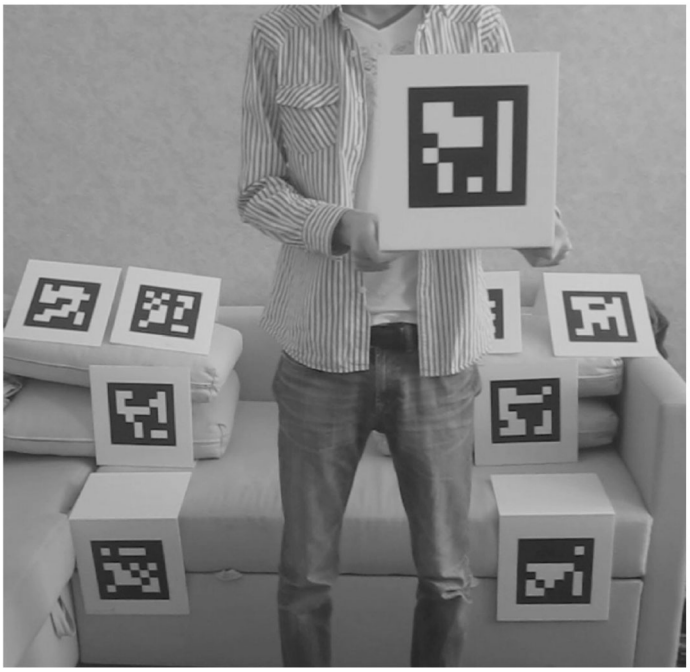

Fig. 9. A frame with a marker representing a moving object.

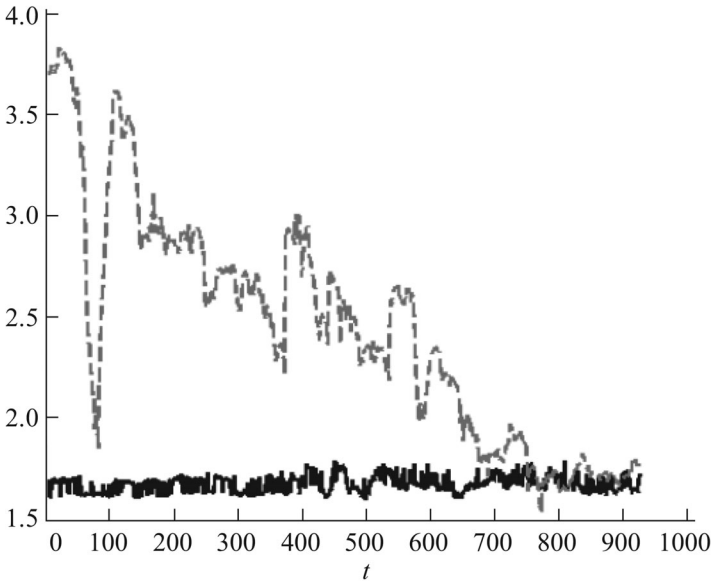

Fig. 10. Trajectories of depth estimates for the point (dashed line) and its true values (solid line) in case when the observed object is moving.

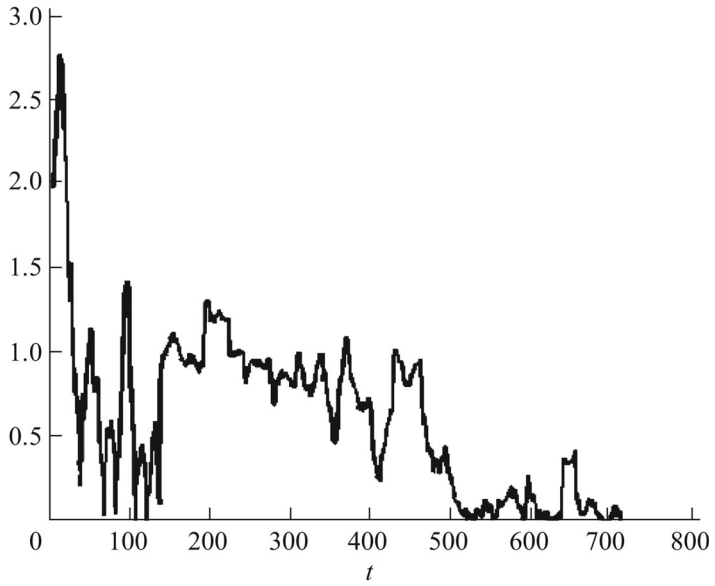

Fig. 11. Error in the point depth estimates in case of a static observed object.

they can be automatically detected with sufficient accuracy on camera images [28]. The location of markers with respect to each other was reconstructed with reconstruction methods by several photos [3], with a high-definition camera (3648 × 2736 pixels) and a tripod to provide sharper images and, accordingly, more exact estimates for the markers' positions. The base of markers can be represented as a set of points in the three-dimensional space (angles of white regions inside a marker) whose coordinates are known. When we have a set of three-dimensional points and a set of their projections, data on the location and orientation of the camera with which these projections were obtained can be computed by standard reconstruction methods [3]. In our experiment, the

observed object was also represented as a marker (Fig. 9). We reconstructed the true object location with reconstruction methods under the assumption that the marker is flat. Photos were taken in daylight, with a camera shooting with frequency 30 Hz and resolution 1280×720 pixels. Two persons participated in the experiment: one held the camera in his hands and performed periodic oscillations in various directions to simulate random motion with zero expectation. The observed object was controlled by the second person positioned at a distance of approximately 1.5 meters from the camera. To estimate the position, the algorithm chose a single point on the marker and tracked its projection not with the marker detector but with optical flow algorithms [29] in order to better approximate real life conditions. We considered two cases: a static object and a moving object. Resulting estimates produced by the algorithm are shown on Figs. 10 and 11, which show that the algorithm's estimates gradually converge to true values, and in case of a motionless object convergence is slightly faster. Our experiments show that the algorithm can be applied in real life conditions and that our assumptions regarding the known motion of the camera and its internal parameters do make sense as well.

7. CONCLUSION

In this work, we presented a new approach to estimating the position of a moving object observed by a single camera. The algorithm is based on randomizing the camera location that lets us reconstruct the object's trajectory even in cases when standard approaches cannot be applied at all. In practice, such a system can be implemented with special devices that control camera position or, if either the camera itself or the object where it is installed are moving with a certain element of randomness, this motion can be reconstructed with the help of static objects in the scene [30], and then we can applied the proposed new estimation algorithm.

In this work, we have shown the theoretical consistency of algorithm estimates and shown the results of its testing under various conditions. Numerical modeling indicates that the method successfully handles the problem of tracking the object's trajectory in all considered scenarios. In future work, to develop the proposed new method we plan to apply to the resulting estimates an approach based on the Kalman filter [31, 32] in order to predict object velocity more accurately judging by given assumptions on the character of its motion.

ACKNOWLEDGMENTS

This work was supported in part by the Russian Foundation for Basic Research, project no. 13-07-00250, and St. Petersburg State University, project no. 6.37.181.2014.

The authors are grateful to the anonymous referees for constructive comments and ideas that have let us significantly improve the presentation of our results.

APPENDIX

Proof of Theorem. In the proof we will use the following auxiliary Lemma 1 from [20].

Lemma 1 [20]. *If* $e_n > 0$, $\alpha, m > 0$, $\alpha\, m < 1$, $b, k \geqslant 0$,

$$e_n \leqslant (1 - \alpha\, m)\, e_{n-1} + 2\, b \sqrt{e_{n-1}} + k, \quad n = 1, 2, \ldots, \tag{A.1}$$

then for every $\varepsilon > 0$ *there exists* N *such that*

$$\forall n > N \quad e_n \leqslant \left(\frac{b + \sqrt{b^2 + \alpha\, m\, k}}{\alpha\, m} \right)^2 + \varepsilon.$$

Proof of Lemma 1 can be found in [20] or [21].

We denote $\nu_n = \hat{\Gamma}_n - \Gamma_n$, $y_n = \frac{\Delta_n^{(1)}}{\sigma_1^2}(p_n^{(1)} - p_{n-1}^{(1)})$, where $E_n\{\cdot\}$ is the conditional expectation with respect to the entire previous history of observations up until time moment $n-1$.

Due to algorithm (4), with the projection property the conditional expectation of the error at time moment n can be represented as follows:

$$\begin{aligned} E_n\left\{\nu_n^2\right\} &\leqslant E_n\left\{(\hat{\Gamma}_{n-1} - \Gamma_{n-1} + \Gamma_{n-1} - \Gamma_n - \alpha g_n)^2\right\} \\ &= \nu_{n-1}^2 + E_n\left\{(\Gamma_{n-1} - \Gamma_n)^2\right\} + 2\nu_{n-1}E_n\left\{\Gamma_{n-1} - \Gamma_n\right\} \\ &\quad -2\alpha\nu_{n-1}E_n\left\{g_n\right\} - 2\alpha E_n\left\{g_n(\Gamma_{n-1} - \Gamma_n) + \alpha^2 g_n^2\right\}. \end{aligned} \tag{A.2}$$

Let us sequentially estimate all terms (except the first) in the right-hand side of inequality (A.2).

Since the drift speed is bounded and the test disturbance is centered and bounded in mean squares, we have

$$E_n\{(\Gamma_{n-1} - \Gamma_n)^2\} = E_n\left\{\frac{(P_n^{(3)} - P_{n-1}^{(3)})^2}{(P_{n-1}^{(3)}P_n^{(3)})^2}\right\} \leqslant \Gamma_{\max}^4(D_3^2 + \sigma_3^2), \tag{A.3}$$

$$2\nu_{n-1}E_n\{\Gamma_{n-1} - \Gamma_n\} = \frac{2}{P_{n-1}^{(3)}}E_n\left\{\frac{P_{n-1}^{(3)} - P_n^{(3)}}{P_n^{(3)}}\right\}\nu_{n-1} \leqslant 2\Gamma_{\max}^2(D_3 + \delta_3)|\nu_{n-1}|. \tag{A.4}$$

Due to the above notation, we can represent the values of y_n as follows:

$$y_n = \frac{\Delta_n^{(1)}}{\sigma_1^2}\left(\frac{P_{n-1}^{(1)} + \Delta_n^{(1)} + \xi_n^{(1)}}{P_{n-1}^{(3)} + \Delta_n^{(3)} + \xi_n^{(3)}} - \frac{P_{n-1}^{(1)}}{P_{n-1}^{(3)}} + v_n^{(1)} - v_{n-1}^{(1)}\right).$$

Since components of the test disturbance are centered, mutually independent and independent of $\xi_n^{(1)}, \xi_n^{(3)}, v_n^{(1)}$, for the conditional expectation $E_n\{y_n\}$ we find that

$$\begin{aligned} E_n\{y_n\} &= E_n\left\{\frac{\Delta_n^{(1)}}{\sigma_1^2}\right\}E_n\left\{\frac{P_{n-1}^{(1)} + \xi_n^{(1)}}{P_{n-1}^{(3)} + \Delta_n^{(3)} + \xi_n^{(3)}} - \frac{P_{n-1}^{(1)}}{P_{n-1}^{(3)}} + v_n^{(1)} - v_{n-1}^{(1)}\right\} \\ &\quad + E_n\left\{\frac{(\Delta_n^{(1)})^2}{\sigma_1^2}\right\}E_n\left\{\frac{1}{P_{n-1}^{(3)} + \Delta_n^{(3)} + \xi_n^{(3)}}\right\} \\ &= 0 \times E_n\left\{\frac{P_{n-1}^{(1)} + \xi_n^{(1)}}{P_{n-1}^{(3)} + \Delta_n^{(3)} + \xi_n^{(3)}} - \frac{P_{n-1}^{(1)}}{P_{n-1}^{(3)}} + v_n^{(1)} - v_{n-1}^{(1)}\right\} + E_n\{\Gamma_n\} = E_n\{\Gamma_n\}. \end{aligned}$$

Consequently, taking into account (A.4) we get

$$\begin{aligned} -2\alpha\nu_{n-1}E_n\{g_n\} &= -2\alpha\nu_{n-1}(\hat{\Gamma}_{n-1} - E_n\{\Gamma_n\}) \\ &= -2\alpha\nu_{n-1}(\hat{\Gamma}_{n-1} - \Gamma_{n-1} + \Gamma_{n-1} - E_n\{\Gamma_n\}) \\ &\leqslant -2\alpha\nu_{n-1}^2 + 2\alpha\Gamma_{\max}^2(D_3 + \delta_3)|\nu_{n-1}|. \end{aligned} \tag{A.5}$$

For the next term in the right-hand side of inequality (A.2), we sequentially derive

$$-2\alpha E_n\{g_n(\Gamma_{n-1}-\Gamma_n)\} = -2\alpha E_n\{(\hat{\Gamma}_{n-1}-y_n)(\Gamma_{n-1}-\Gamma_n)\}$$
$$= -2\alpha\Gamma_{n-1}\hat{\Gamma}_{n-1} - 2\alpha E_n\{y_n\Gamma_n\} + 2\alpha\hat{\Gamma}_{n-1}E_n\{\Gamma_n\} + 2\alpha\Gamma_{n-1}E_n\{\Gamma_n\}$$
$$= \alpha\nu_{n-1}^2 + 2\alpha E_n\{\Gamma_n^2\} - 2\alpha E_n\{y_n\Gamma_n\} - \alpha E_n\{(\hat{\Gamma}_{n-1}-\Gamma_n)^2\}$$
$$- \alpha E_n\{(\Gamma_{n-1}-\Gamma_n)^2\} \leqslant \alpha\nu_{n-1}^2 + 2\alpha\Gamma_{\max}^2, \tag{A.6}$$

since $E_n\{y_n\Gamma_n\} = E_n\{\Gamma_n^2\} > 0$ because components of the test disturbance are centered, mutually independent and independent of $\xi_n^{(1)}, \xi_n^{(3)}, v_n^{(1)}$.

For the last term in the right-hand side of inequality (A.2), taking into account (A.3) and (A.4) we get

$$\alpha^2 E_n\{g_n^2\} = \alpha^2 E_n\left\{(\hat{\Gamma}_{n-1}-\Gamma_{n-1}+\Gamma_{n-1}-y_n)^2\right\}$$
$$= \alpha^2\nu_{n-1}^2 + 2\alpha^2\nu_{n-1}(\Gamma_{n-1}-E_n\{\Gamma_n\})$$
$$+ \alpha^2\left(\Gamma_{n-1}^2 - 2\Gamma_{n-1}E_n\{\Gamma_n\} + E_n\{\Gamma_n^2\} + E_n\{y_n^2-\Gamma_n^2\}\right)$$
$$\leqslant \alpha^2\nu_{n-1}^2 + 2\alpha^2\Gamma_{\max}^2(D_3+\delta_3)|\nu_{n-1}|$$
$$+ \alpha^2\Gamma_{\max}^4(D_3^2+\sigma_3^2) + \alpha^2\left(\frac{\sigma_2^2}{\sigma_1^4}(2H_1+2c_d)^2 + \frac{\sigma_4^4}{\sigma_1^4}\Gamma_{\max}^2\right), \tag{A.7}$$

and since components of the test disturbance are centered, mutually independent, and independent of $\xi_n^{(1)}, \xi_n^{(3)}, v_n^{(1)}$ we get

$$E_n\left\{y_n^2-\Gamma_n^2\right\} < E_n\left\{y_n^2\right\}$$
$$= E_n\left\{\left(\frac{\Delta_n^{(1)}}{\sigma_1^2}\left(\frac{P_{n-1}^{(1)}+\xi_n^{(1)}}{P_n^{(3)}} - \frac{P_{n-1}^{(1)}}{P_{n-1}^{(3)}} + v_n^{(1)} - v_{n-1}^{(1)}\right)\right)^2\right\}$$
$$+ E_n\left\{\left(\frac{(\Delta_n^{(1)})^2}{\sigma_1^2 P_n^{(3)}}\right)^2\right\} \leqslant \frac{\sigma_2^2}{\sigma_1^4}(2H_1+2c_d)^2 + \frac{\sigma_4^4}{\sigma_1^4}\Gamma_{\max}^2.$$

As a result, substituting into (A.2) the resulting estimates (A.3)–(A.7) we conclude that

$$E_n\{\nu_n^2\} \leqslant (1-\alpha+\alpha^2)\nu_{n-1}^2 + 2(1+\alpha+\alpha^2)\Gamma_{\max}^2(D_3+\delta_3)|\nu_{n-1}|$$
$$+ (1+\alpha^2)\Gamma_{\max}^4(D_3^2+\sigma_3^2) + \alpha\Gamma_{\max}^2\left(2+\alpha\left(4\Gamma_{\min}^{-1} + 2D_1 + 2c_d^2 + \frac{\sigma_4^4}{\sigma_1^4}\right)\right)$$
$$= (1-\alpha(1-2\alpha))\nu_{n-1}^2 + 2\,b\,|\nu_{n-1}| + k. \tag{A.8}$$

Taking unconditional expectation of both sides in (A.8), we see that all conditions Lemma 1 hold for $m = 1-\alpha$, and $e_n = E\nu_n^2$ since $(E|\nu_n|)^2 \leqslant E\nu_n^2$. Applying Lemma 1, we get formula (6), the first item of theorem.

Formulas (7) can be derived from estimates on the differences

$$E|\hat{P}_n^{(i)} - P_n^{(i)}| = E\left|\frac{p_n^{(i)}}{\hat{\Gamma}_n} - (p_n^{(i)} - v_n^{(i)})P_n^{(3)}\right|$$
$$\leqslant E\left|p_n^{(i)}\left(\frac{1}{\hat{\Gamma}_n} - \frac{1}{\Gamma_n}\right)\right| + |v_n^{(i)}P_n^{(3)}|$$
$$\leqslant \Gamma_{\min}^{-2}H_iE|\nu_n| + \Gamma_{\min}^{-1}c_i, \quad i = 1, 2.$$

This concludes the proof of theorem.

REFERENCES

1. Lucas, B.D. and Kanade, T., An Iterative Image Registration Technique with an Application to Stereo Vision, *Proc. Int. Joint Conf. Artific. Intelligence*, Vancouver, 1981, vol. 81, pp. 674–679.
2. Klein, G. and Murray, D., Parallel Tracking and Mapping for Small AR Workspaces, *Proc. 6th IEEE ACM Int. Sympos. Mixed Augmented Reality*, Nara, Japan, 2007, pp. 225–234.
3. Hartley, R. and Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge: Cambridge Univ. Press, 2003.
4. Zhengyou, Z., Microsoft Kinect Sensor and Its Effect, *IEEE MultiMedia*, 2012, vol. 19, no. 2, pp. 4–10.
5. Premebida, C., Monteiro, G., Nunes, U., and Peixoto, P., A Lidar and Vision-Based Approach for Pedestrian and Vehicle Detection and Tracking, *Proc. IEEE Intelligent Transportat. Syst. Conf.*, Bellevue, 2007, pp. 1044–1049.
6. Sturm, P., Structure and Motion for Dynamic Scenes. The Case of Points Moving in Planes, *Proc. Eur. Conf. Comput. Vision*, Copenhagen, 2002, pp. 507–509.
7. Han, M. and Kanade, T., Reconstruction of a Scene with Multiple Linearly Moving Objects, *Proc. IEEE Conf. Comput. Vision Patt. Recognit.*, Hilton Head Island, 2000, pp. 542–549.
8. Krivokon', D.S. and Vakhitov, A.T., Randomization in the Problem of Estimating the Depth of a Point with a Single Camera Based on the Idea of an Asymptotic Observer, *Stokh. Optim. Informat.*, 2012, vol. 8, no. 2, pp. 49–59.
9. Krivokon, D. and Vakhitov, A., Randomized Algorithm for Estimation of Moving Point Position Using Single Camera, *Proc. 53rd IEEE Conf. Decision Control*, 2014, pp. 5189–5194.
10. Wedel, A., Franke, U., Klappstein, J., Brox, T., and Cremers, D., Realtime Depth Estimation and Obstacle Detection from Monocular Video, *Patt. Recognit., Lecture Notes Comput. Sci.*, 2006, vol. 4174, pp. 475–484.
11. Granichin, O.N., A Stochastic Recursive Procedure with Correlated Noises in the Observation, that Employs Trial Perturbations at the Input, *Vest. Leningr. Univ.: Mat.*, 1989, vol. 22, no. 1, pp. 27–31.
12. Granichin, O.N., Procedure of Stochastic Approximation with Disturbances at the Input, *Autom. Remote Control*, 1992, vol. 53, no. 2, part 1, pp. 232–237.
13. Granichin, O.N., Estimating the Minimum Point for an Unknown Function Observed with Dependent Noise, *Probl. Peredachi Inf.*, 1992, no. 2, pp. 16–20.
14. Polyak, B.T. and Tsybakov, A.B., Optimal Orders of Precision for Stochastic Approximation Search Algorithms, *Probl. Peredachi Inf.*, 1990, no. 2, pp. 45–53.
15. Spall, J.C., Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation, *IEEE Trans. Automat. Control*, 1992, vol. 37, no. 3, pp. 332–341.
16. Kushner, H.J. and Yin, G.G., *Stochastic Approximation Algorithms and Applications*, New York: Springer-Verlag, 2002.

17. Borkar, V.S., *Stochastic Approximation: A Dynamical Systems Viewpoint*, Cambridge: Cambridge Univ. Press, 2008.

18. Granichin, O., Volkovich, V., and Toledano-Kitai, D., *Randomized Algorithms in Automatic Control and Data Mining*, London: Springer-Verlag, 2015.

19. Vakhitov, A.T., Granichin, O.N., and Gurevich, L.S., Algorithm for Stochastic Approximation with Trial Input Perturbation in the Nonstationary Problem of Optimization, *Autom. Remote Control*, 2009, vol. 70, no. 11, pp. 1827–1835.

20. Granichin, O., Gurevich, L., and Vakhitov, A., Discrete-Time Minimum Tracking Based on Stochastic Approximation Algorithm with Randomized Differences, *48th Conf. Decision Control*, Shanghai, China, 2009, pp. 5763–5767.

21. Granichin, O.N., Stochastic Approximation Search Algorithms with Randomization at the Input, *Autom. Remote Control*, 2015, vol. 76, no. 5, pp. 762–775.

22. Amelin, K.S. and Granichin, O.N., Possibilities for Randomization in Prediction Algorithms of Kalman Type for Arbitrary External Noise in Observation, *Giroskop. Navigat.*, 2011, no. 2, pp. 38–50.

23. Granichin, O.N. and Amelina, N.O., Simultaneous Perturbation Stochastic Approximation for Tracking under Unknown but Bounded Disturbances, *IEEE Trans. Automat. Control*, 2015, vol. 60, no. 6, pp. 1653–1658.

24. Brown, D., Decentering Distortion of Lenses, *Photometric Eng.*, 1966, vol. 32, no. 3, pp. 444–462.

25. Escalera, A. and Armingol, J., Automatic Chessboard Detection for Intrinsic and Extrinsic Camera Parameter Calibration, *Sensors*, 2010, vol. 10, no. 3, pp. 2027–2044.

26. Zhang, Z., Flexible Camera Calibration by Viewing a Plane from Unknown Orientations, *Proc. 7 IEEE Int. Conf. Comput. Vision*, 1999, vol. 1, pp. 666–673.

27. Brown, M. and Lowe, D., Automatic Panoramic Image Stitching Using Invariant Features, *Int. J. Comput. Vision*, 2007, vol. 74, no. 1, pp. 59–73.

28. Fiala, M., ARTag, a Fiducial Marker System Using Digital Techniques, *Proc. IEEE Comput. Soc. Conf. Comput. Vision Patt. Recognit.*, 2005, vol. 2, pp. 590–596.

29. Barron, J., Fleet, D., and Beauchemin, S., Performance of Optical Flow Techniques, *Int. J. Comput. Vision*, 1994, vol. 12, no. 1, pp. 43–77.

30. Arsen'ev, D.G., Ivanov, V.M., and Berkovskii, N.A., Navigation by Distances to Reference Points with an Adaptive Method of Significant Sampling, *Nauchn.-tekhn. vedomosti SPbGPU*, Ser. *Informat. Telekommun. Upravlenie*, 2011, no. 1 (115), pp. 81–86.

31. Stepanov, O.A., Kalman Filter: History and State of the Art (To the 80th Birthday of Rudolf Kalman), *Giroskop. Navigats.*, 2010, no. 2, pp. 107–120.

32. Stepanov, O.A. and Toropov, A.B., Linear Optimal Algorithms in Estimation Problems with Nonlinear Measurements. Relation to Kalman Type Algorithms, *Izv. Tulsk. Gos. Univ., Tekhn. Nauki*, 2012, no. 7, pp. 172–190.

This paper was recommended for publication by O.A. Stepanov, a member of the Editorial Board