Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0109.v1

Article

Moving object detection based on a combination of Kalman filter and median filtering

Diana Kalita[1], **Pavel Lyakhov**[1,2]

[1] Department of Mathematical Modeling, North Caucasus Federal University, 355017 Stavropol, Russia; diana.kalita@mail.ru
[2] Department of Modular Computing and Artificial Intelligence, North-Caucasus Center for Mathematical Research, 355017 Stavropol, Russia; ljahov@mail.ru
* Correspondence: diana.kalita@mail.ru Tel.: +7-962-007-90-24

Abstract: The task of determining the distance from one object to another is one of the important tasks solved in robotics systems. Conventional algorithms rely on an iterative process of predicting distance estimates, which results in an increased computational burden. Algorithms used in robotic systems should require minimal time costs, as well as be resistant to the presence of noise. To solve these problems, the paper proposes an algorithm for Kalman combination filtering with a Goldschmidt divisor and a median filter. Software simulation showed an increase in the accuracy of predicting the estimate of the developed algorithm in comparison with the traditional filtering algorithm, as well as an increase in the speed of the algorithm. The results obtained can be effectively applied in various computer vision systems.

Keywords: Kalman filter, median filter, impulse noise, estimate prediction, object distance determination, lidar, value calibration, point cloud.

1. Introduction

Currently, solving problems related to determining the distance to an object is practically significant in various robotic systems, including computer vision systems [1]. Modern computer vision systems obtain real information about the distance to objects using various sets of sensors, such as radar [2], ultrasonic sensors [3], lidars [4], stereo mate [5], etc.

At the same time, in various applications (unmanned vehicles, mobile applications), the approach of merging lidar and camera sensors is becoming more widespread [6]. In [6], the authors propose an approach to merge several lidar sensors and a color camera in real time to recognize multi-scale objects at the semantic level, which makes it possible to adapt the developed module to complex scenes.

The procedure for the simultaneous use of such sensors is that the common sensor receives a combination of data coming from the camera in the form of an image, and from the lidar in the form of a cloud of points. After that, the resulting point cloud is projected onto the image. The position of the point cloud in the image is the determined distance to the moving object. In [7], the authors propose to determine the distance to objects of known sizes and geometry in order to improve the estimation of the position of the object before quantization and to obtain systematic errors inherent in lidar. The authors also propose in their work to apply the fitting method to convert the lidar into a monocular camera, which avoids the task of extracting the target edge from the point cloud. For successful sensor fusion, [8] uses sparse and accurate point clouds as a guideline for determining the correspondence of stereo images in a single three-dimensional volume space. In [9], a lidar-camera fusion approach was developed to correct oscillating lidar distortions with full velocity estimation. A new mechanism for quantifying and analyzing 3D

© 2022 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0109.v1

motion correlation for estimating real-time temporal displacement of fusion of dissimilar sensors is described by the authors of [10]. A multi-sensor platform for combining camera and lidar data for object detection, including small obstacles and moving objects, is proposed in [11]. A new merge pipeline based on the method of early fusion of image range and RGB image to improve the detection of 3-D objects is proposed in [12]. A new approach to the use of a convolutional neural network that detects and identifies an object based on data obtained from a 3-D lidar is described in [13]. Paper [14] presents a new method for estimating the transformation between the manipulator camera and the 2D lidar coordinate system, based on the use of point, line, and plane geometry constraints between the segmented 2D lidar scan and the reconstructed trihedron elements. The proposed spatiotemporal sampling algorithm activates the lidar only in areas of interest identified by analyzing visual input and reduces the base lidar frame rate according to the kinematic state of the system in [15]. In [16], the authors, before calibrating the lidar camera, use a new method in which they align 3D visual points on laser scans based on the tight coupling graph optimization method to calculate the external parameter between the lidar and the camera. The approach [17] uses the geometric information provided by the point cloud as prior knowledge and clusters the point cloud data using an improved density clustering algorithm to further combine data from lidar and camera sensors. The authors of [18] in their work at the theoretical level analyze the limitations imposed by boundary objects and the sensitivity of the calibration accuracy to the distribution of boundaries in the scene, and at the implementation level they propose a method for detecting lidar boundaries based on cutting voxels from a point cloud and fitting the plane. Three different approaches for combining image characteristics with point cloud data in which lidar reflection data can be replaced with low-level image characteristics without degrading detector performance have been developed by researchers [19]. To solve the problem of combining data from different types of lidar and camera sensors, the authors of [20] developed a multimodal system for detecting and tracking 3D objects (EZFusion). The classification of 3D LiDAR point clouds using a visual object detector and an object tracker that jointly performs detection and tracking of 3D objects before data calibration is used in [21].

However, in all the listed works, the sensor data are calibrated without their preliminary processing. Since there may be impulsive noise among the data during projection, then in this case, in order to improve the accuracy of distance estimation, the received data require primary processing with a digital filter. From this point of view, a suitable tool is the median filter [22], which allows cleaning data from impulse noise. On the other hand, the process of combining several sensors at the same time requires the use of a suitable algorithm that allows you to work with multiple input signals. The Kalman filter [23] is a recursive filter capable of predicting the future state of a system based on previous data.

However, in real-time computer vision systems, the process of estimating the distance to a moving object should be as fast as possible. The presence of the division operation in the traditional Kalman filter algorithm increases the computational cost [24]. There are various approaches that allow to increase the speed of a computing system built on the basis of the Kalman filter, among which are the works [24-26] and others. In this paper, we propose an algorithm that includes combinational filtering of data based on a median filter used to filter a point cloud and a modified Kalman filter with a Goldschmidt divider used to predict estimates of distances to a moving object.

The article includes: Section 1 containing preliminary information and problem statement; Section 2, which introduces the developed filter algorithm; Section 3 describes software and experimental simulations of filter accuracy and time performance estimates.

2. Materials and Methods

The Kalman filter is a recursive probabilistic filter. Based on the filter, the past state, present state and future state of the dynamic system are determined [27]. In this case, the entire filtering procedure can be divided into two processes, consisting in performing

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0109.v1

prediction and updating the state. The predicted state is a mathematical model (1) and (2). They use previous and input data received from a series of sensors.

$$\bar{x}_k = Ax_{k-1} + Bu_{k-1} \tag{1}$$

where $\bar{x}_k$ – state variable vector, A – transition matrix between system states, B – control matrix, u_{k-1} – system control action at the previous moment of time.

$$\bar{P}_k = AP_{k-1}A^T + Q \tag{2}$$

where $\bar{P}_k$ – error covariance matrix, Q – technological noise covariance matrix.

The updated state uses the predicted state value to perform the next prediction step. The updated state is calculated using equations (4) and (5). At each stage of the filtering, both the prediction and update of the state and the prediction and update of the covariance error are calculated.

$$K_k = \frac{\bar{P}_{k-1}}{H\bar{P}_{k-1}H^T + R} \tag{3}$$

$$\hat{x}_k = \bar{x}_{k-1} + K_k(y_k - H\bar{x}_{k-1}) \tag{4}$$

$$\hat{P}_k = (I - K_kH)\bar{P}_{k-1} \tag{5}$$

where K_k – Kalman gain, H – measurement matrix showing the relationship between measurements and the state of the system, R – measurement noise matrix, y_k – measurement of the system state at the current time, I – identity matrix.

However, the step of updating the states of a dynamical system requires a preliminary determination of the value of the Kalman gain (3) [28] . Since Kalman gain involves performing a division operation, it can be concluded that gain calculation is the most computationally expensive operation.

There are many methods for designing fission equipment. Among which we can single out the quadratically convergent Goldschmidt algorithm [29]. A feature of this algorithm is the ability to obtain the result of division without a remainder. For exact rounding of the quotient, it is usually necessary to use an additional multiplication of the quotient and the divisor, and then subtract the product from the dividend to get the final remainder.

Let’s represent the quotient as:

$$N = \frac{a}{b} \tag{6}$$

Then this algorithm is defined by the expression (7):

$$N = \frac{a * F_i}{b * F_i} \tag{7}$$

where coefficient $F_i = 2 - b$. As a result, the iterative multiplication of the dividend and the divisor of the fraction makes it possible to bring the denominator of the fraction to 1, and take the resulting numerator as an approximate division value. In section 2, we will show the application of the Goldschmidt algorithm in the Kalman filter when calculating the gain K_k, and we will also use the median filter to filter the point cloud projected onto the input image.

The median filter is a non-linear digital filter [30]. Such a filter is used in image and signal processing to remove impulse noise and smooth the signal. In systems that receive information using sensors, the signals always have noise from the environment and noise from the loss of sensors.

The process of median filtering is to obtain the signal values in the form of a variational series built in ascending or descending order. Signal values are taken in the vicinity of some point (filter window). To solve the task of determining the distance to a moving object, the median filter will select a cloud of points from the input images using nearby data. Thus, if we designate the filtered selection of array elements as $D = \{d_1, d_2, \dots, d_n\}$, so that the number of selection elements coincides with the size of the filter window, then the application of median filtering, which selects the central values of the ordered selection, can be written as:

$$x^* = med\left(d_1, d_2, \dots, d_n\right) \tag{8}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0109.v1

3. Results

This section presents a filtering algorithm that uses the median filtering method, as well as a modified Kalman filter based on the Goldschmidt divisor.

3.1. Proposed algorithm

Let some point $m = [u, v]^T$ on the image plane be given. Moreover, it is a projection of some point $M = [X, Y, Z]^T$ in the space of the point cloud. Then the relationship between these points can be written as:

$$S * m = W * [\, R^* t] * M \tag{9}$$

where S – some scalar, R^* and t – external parameters of the camera, W – matrix of internal parameters of the camera (10):

$$W = \begin{bmatrix} \alpha & \gamma & x_o \\ 0 & \beta & y_o \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

where (x_0, y_0) – coordinates of the point, α and β – compression ratios along the Ox and Oy axes, , γ – coefficient of asymmetry between the image axes. We take $Z = 0$ and denote by r_{ij} the i-th element of the rotation matrix R^*. Then the projective matrix M_p (11), which projects the point cloud onto the image obtained by calibrating the lidar and camera sensors, will take the form:

$$M_p = \begin{bmatrix} r_{11} & r_{12} & x_0 \\ r_{21} & r_{22} & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

Then we rewrite (9) for a point on the object $M^* = [X, Y, 1]^T$ and the corresponding point on the image m^* in the form:

$$S * m^* = L * M^* \tag{12}$$

where L – image homography matrix, defined as:

$$L = W[r_1 \quad r_2 \quad t] \tag{13}$$

Let us find the homography matrix, for this we introduce the covariance matrix Φ_{m_i}

$$\textstyle\sum_i (m_i - m_i^*)^T \Phi_{m_i}^{-1} (m_i - m_i^*) \tag{14}$$

where m_i^* determine from the expression (15):

$$m_i^* = \frac{1}{\bar{l}_3^T M_i} \begin{bmatrix} \bar{l}_1^T M_i \\ \bar{l}_2^T M_i \end{bmatrix} \tag{15}$$

where $\bar{l}_i$ – i-th row of the matrix L.

Let $L = [l_1 \quad l_2 \quad l_3]$, then from (12) we get:

$$[l_1 \quad l_2 \quad l_3] = \lambda W [r_1 \quad r_2 \quad t]$$

where λ – some scalar. Knowing that r_1 and r_2 are orthogonal, then we write:

$$l_1^T W^{-T} W^{-1} l_2 = 0 \tag{16}$$

$$l_1^T W^{-T} W^{-1} l_1 = l_2^T W^{-T} W^{-1} l_2 \tag{17}$$

For v_{ij} we get:

$$v_{ij} = \left[l_{i1} l_{j1}, l_{i1} l_{j2} + l_{i2} l_{j1}, l_{i2} l_{j2} + l_{i3} l_{j1} + l_{i1} l_{j3}, l_{i3} l_{j2} + l_{i2} l_{j3}, l_{i3} l_{j3}\right]^T$$

Then we rewrite (16) and (17) as:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} e = 0 \tag{18}$$

Where is the vector e:

$$e = [E_{11}, E_{12}, E_{22}, E_{13}, E_{23}, E_{33}\,]$$

E_{ij} matrix elements $E = W^{-T} W^{-1}$. Then, to determine the matrix W of the internal parameters of the camera, it suffices to use n images and equations (19):

$$v_0 = (E_{12} E_{13} - E_{11} E_{23}) / (E_{11} E_{22} - E_{12}^2)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0109.v1

$$\lambda = E_{33} - [E_{13}^2 + v_0(E_{12}E_{13} - E_{11}E_{23})]/E_{11}$$

$$\alpha = \sqrt{\lambda/E_{11}}$$

$$\beta = \sqrt{\lambda E_{11}/(E_{11}E_{22} - E_{12}^2)}$$

$$\gamma = -\frac{E_{12}\alpha^2\beta}{\lambda}$$

$$u_0 = \frac{cv_0}{\alpha} - \frac{E_{13}\alpha^2}{\lambda}$$

After determining the matrix W, it is possible to determine the homography matrix, and using expression (12)) to carry out a complete projection of the point cloud into the image space. Denote $d_{ij} = L * M^*$. Then after projecting the points, we get some matrix of values:

$$D = \begin{bmatrix} d_{11} & \cdots & d_{1j} \\ \vdots & \ddots & \vdots \\ d_{i1} & \cdots & d_{ij} \end{bmatrix} \tag{20}$$

To the resulting array of values, we will apply a median filter of size $N * N$ for pre-processing and removal of impulse noise. This approach will improve the accuracy of the position of the point cloud to further determine the distance to the object. After receiving the processed data, a digital Kalman filter is applied to them. The Kalman filter can predict future status based on previous data. We propose a modified Kalman filter algorithm based on the use of the Goldschmidt divider to calculate the Kalman gain value.

Thus, based on the conditions described, the paper proposes an algorithm for implementing data filtering to detect the distance to an object in a video data stream

Proposed algorithm: Data filtering using median filter and Kalman filter built using the Goldschmidt algorithm to calculate the Kalman gain

```
Input data:
1:    ${m, M, S, R^*, L, E, r_1, r_2, t, x_{0,i}, P_{o,i}, A, B, C, D, H, Q, R}$
Calibration of sensor values:
Determination of the internal parameters of the camera matrix $W$:
2.    $v_0 = (E_{12}E_{13} - E_{11}E_{23})/(E_{11}E_{22} - E_{12}^2)$
3.    $\lambda = E_{33} - [E_{13}^2 + v_0(E_{12}E_{13} - E_{11}E_{23})]/E_{11}$
4.    $\alpha = \sqrt{\lambda/E_{11}}$
5.    $\beta = \sqrt{\lambda E_{11}/(E_{11}E_{22} - E_{12}^2)}$
6.    $\gamma = -\frac{E_{12}\alpha^2\beta}{\lambda}$
7.    $u_0 = \frac{cv_0}{\alpha} - \frac{E_{13}\alpha^2}{\lambda}$
Definition of homography matrix:
8.    $L = W[r_1 \quad r_2 \quad t]$
Projection matrix calculation $D$:
9.    $d_{ij} = L * M^*$
Calculation of median filtering values:
10.   $x^* = med(d_1, d_2, \dots, d_n)$
Kalman filtering:
Prediction Calculation:
11:   $\bar{x}_k = Ax_{k-1} + Bu_{k-1}$
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0109.v1

```
12:     $\bar{P}_k = AP_{k-1}A^T + Q$
Kalman gain calculation:
13:     $b_0 = H\bar{P}_k H^T + R$
14:     $F_k = 2 - b_k$
15:     $N_k = N_{k-1}F_k$
16:     $b_k = b_{k-1}F_k$
    If $b_k = 1$, then
17:     $K_k = \bar{P}_k H^T b_k$
    other $F_{k+1} = 2 - b_k,$
Update Calculation:
18:     $\hat{x}_k = \bar{x}_{k-1} + K_k(y_k - H\bar{x}_{k-1})$
19:     $\hat{P}_k = (I - K_k H)\bar{P}_{k-1}$
```

The developed algorithm makes it possible to obtain such a filter design that is capable of processing data obtained after calibrating the values coming from different types of sensors. At the same time, the use of median filtering preliminarily clears the projected data from different planes from impulse noise. In turn, the modified Kalman filter, due to the use of the Goldschmidt algorithm in calculating the Kalman gain, makes it possible to avoid the division operation in the filter design, which will reduce computational costs during filtering.

Let us consider the numerical implementation of the algorithm described above, which proves the effectiveness of the proposed algorithm and theoretical results in comparison with the known algorithm

3.2. Numerical and software implementation

Let the system under consideration consist of 2 sensors: a lidar sensor and a camera from which images are received. The system matrices and input data are proposed to be equal: $m = [4,3]^T$, $M = [2{,}1{,}0]^T$, $Wi = 10 \times 10$; $\alpha = \beta = 1$, $\gamma = 0$, $R^* = 1$, $t = 1$, $\Delta T = 0{,}05$ – sampling time for each frame, $B = 0$, $\tau = 0{,}05$ – measurement noise variance

$$A = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q = \begin{bmatrix} \Delta T^4/4 & 0 & \Delta T^3/2 & 0 \\ 0 & \Delta T^4/4 & 0 & \Delta T^3/2 \\ \Delta T^3/2 & 0 & \Delta T^2 & 0 \\ 0 & \Delta T^3/2 & 0 & \Delta T^2 \end{bmatrix} * \tau^2$$

$$R = \begin{bmatrix} \tau_1 & 0 & 0 & 0 \\ 0 & \tau_1 & 0 & 0 \\ 0 & 0 & \tau_1 & 0 \\ 0 & 0 & 0 & \tau_1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

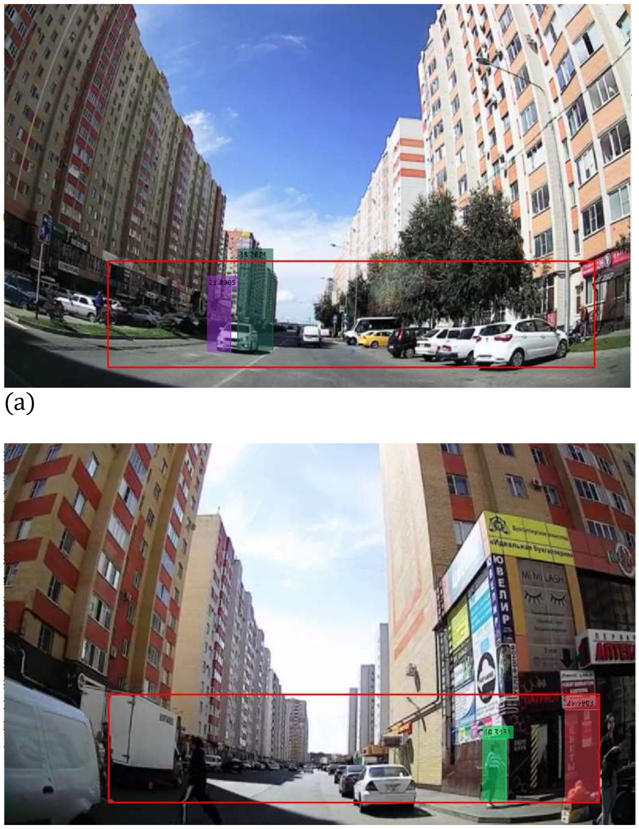
Comparison of the simulation results for the well-known Kalman filtering algorithm and the developed algorithm are presented in Table 1. The error in determining the distance was found as the difference between the true value of the distance and the determined one. Each scenario number has a different distance. Simulation was carried out on 10 scenarios with different true distance values, respectively, from 1 to 10 m.

Table 1. Results in the measurement error and the time of determining the distance to the object in the known and proposed algorithm.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0109.v1

№	**Known Algorithm**		**Proposed Algorithm**	
	Error	Temporary delay	Error	Temporary delay
1	0.89	0.17	0.48	0.12
2	0.65	0.17	0.57	0.14
3	0.87	0.16	0.32	0.14
4	0.35	0.18	0.13	0.16
5	0.78	0.19	0.07	0.15
6	0.26	0.23	0.18	0.19
7	0.37	0.22	0.33	0.19
8	0.58	0.27	0.31	0.22
9	0.82	0.25	0.55	0.25
10	0.82	0.25	0.39	0.22

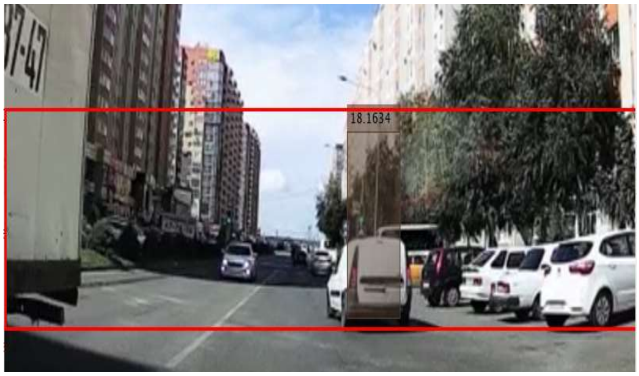
Figure 1 shows the results of software simulation. To carry out software simulation, archival data was used in the form of a video file with a frame size of 720 × 1280 and an input rate of up to 30 frames per second. The simulation was carried out using the Matlab R2021a software package. The purpose of the simulation is to measure the distance from the sensor to the object. The object of interest is in the bounding box, and the bounding box is created by detecting the object. The determined distance corresponds to the measured distance from the point cloud. The data is unstable due to the large number of point clouds from the environment and objects in the bounding box. There was a difficulty in choosing an objective point cloud. The median filter helps you select a bounding box point cloud using nearby data.



(a)

(b)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0109.v1



(c)

Figure 1. The result of software simulation of determining the distance from the object to the car a) the distance to the approaching object; b) distance to turning objects; c) distance to the receding object

4. Discussion

The obtained results show that the developed algorithm gives a gain in the measurement error due to the use of the median filter and the detection time of the distance to the object in the video data stream. Moreover, the greater the distance to the object, the higher the error and the time of determining the distance.

As a result of the program, the distance to moving objects is determined. During the entire video sequence, moving objects can be covered by other objects, projective transformations associated with camera tilt do not occur. However, there are affine transformations associated with scalability and rotation of moving objects. The results obtained make it possible to determine the distance to moving objects when this object is overlapped in computer vision systems.

5. Conclusions

In this paper, we studied the solution of the problem of determining the distance to an object using the well-known Kalman filtering algorithm and the developed filtering algorithm, which uses the Goldschmidt divider design to improve the performance of the computing system. Also, the developed algorithm includes a block of median data filtering to improve the accuracy of determining the distance to the object. The results of software simulation showed a faster determination of the distance to an object in the video data stream, as well as a smaller error in such determination. The results obtained showed that the developed algorithm can be applied in computer vision systems to combine data obtained from different types of sensors. In addition, the proposed algorithm should be applied in computer vision systems, where a critical indicator is to increase the performance of the system, i.e. indicator of the time delay of the system.

Further research will be aimed at developing the hardware implementation of the developed algorithm, the architectural implementation of the median filtering units, and the modified algorithm divider, as well as comparing the work of the developed architectures in positional and non-positional number systems.

Author Contributions: Conceptualization, D.K. and P.L.; methodology, P.L.; software, D.K.; validation, D.K.; formal analysis, P.L.; investigation, D.K.; resources, P.L.; data curation, D.K.; writing—original draft preparation, D.K.; writing—review and editing, P.L.; visualization, D.K.; supervision, P.L.; project administration, P.L.; funding acquisition, D.K. and P.L. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the Council for Grants of the President of the Russian Federation under Project MK-3918.2021.1.6.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0109.v1

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank North-Caucasus Federal University for supporting in the contest of projects competition of scientific groups and individual scientists of North-Caucasus Federal University.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Rahmadya, B.; Sun, R.; Takeda, S.; Kagoshima, K.; Umehira, M. A Framework to Determine Secure Distances for Either Drones or Robots Based Inventory Management Systems. *IEEE Access* **2020**, *8*, 170153–170161, doi:10.1109/ACCESS.2020.3024963.
2. Lee, T.-Y.; Skvortsov, V.; Kim, M.-S.; Han, S.-H.; Ka, M.-H. Application of Band FMCW Radar for Road Curvature Estimation in Poor Visibility Conditions. *IEEE Sens J* **2018**, *18*, 5300–5312, doi:10.1109/JSEN.2018.2837875.
3. Wang, L.; Wang, T.; Liu, H.; Hu, L.; Han, Z.; Liu, W.; Guo, N.; Qi, Y.; Xu, Y. An Automated Calibration Method of Ultrasonic Probe Based on Coherent Point Drift Algorithm. *IEEE Access* **2018**, *6*, 8657–8665, doi:10.1109/ACCESS.2018.2791582.
4. Balemans, N.; Hellinckx, P.; Steckel, J. Predicting LiDAR Data From Sonar Images. *IEEE Access* **2021**, *9*, 57897–57906, doi:10.1109/ACCESS.2021.3072551.
5. Toth, M.; Stojcsics, D.; Domozi, Z.; Lovas, I. Stereo Odometry Based Realtime 3D Reconstruction. In Proceedings of the 2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY); *IEEE*, September 2018; pp. 000321–000326.
6. Deng, Q.; Li, X.; Ni, P.; Li, H.; Zheng, Z. Enet-CRF-Lidar: Lidar and Camera Fusion for Multi-Scale Object Recognition. *IEEE Access* **2019**, *7*, 174335–174344, doi:10.1109/ACCESS.2019.2956492.
7. Huang, J.-K.; Grizzle, J.W. Improvements to Target-Based 3D LiDAR to Camera Calibration. *IEEE Access* **2020**, *8*, 134101–134110, doi:10.1109/ACCESS.2020.3010734.
8. Choe, J.; Joo, K.; Imtiaz, T.; Kweon, I.S. Volumetric Propagation Network: Stereo-LiDAR Fusion for Long-Range Depth Estimation. *IEEE Robot Autom Lett* **2021**, *6*, 4672–4679, doi:10.1109/LRA.2021.3068712.
9. Yang, W.; Gong, Z.; Huang, B.; Hong, X. Lidar With Velocity: Correcting Moving Objects Point Cloud Distortion From Oscillating Scanning Lidars by Fusion With Camera. *IEEE Robot Autom Lett* **2022**, *7*, 8241–8248, doi:10.1109/LRA.2022.3187506.
10. Qiu, K.; Qin, T.; Pan, J.; Liu, S.; Shen, S. Real-Time Temporal and Rotational Calibration of Heterogeneous Sensors Using Motion Correlation Analysis. *IEEE Transactions on Robotics* **2021**, *37*, 587–602, doi:10.1109/TRO.2020.3033698.
11. Zhangyu, W.; Guizhen, Y.; Xinkai, W.; Haoran, L.; Da, L. A Camera and LiDAR Data Fusion Method for Railway Object Detection. *IEEE Sens J* **2021**, *21*, 13442–13454, doi:10.1109/JSEN.2021.3066714.
12. Zhang, Z.; Liang, Z.; Zhang, M.; Zhao, X.; Li, H.; Yang, M.; Tan, W.; Pu, S. RangeLVDet: Boosting 3D Object Detection in LIDAR With Range Image and RGB Image. *IEEE Sens J* **2022**, *22*, 1391–1403, doi:10.1109/JSEN.2021.3127626.
13. Zhao, X.; Sun, P.; Xu, Z.; Min, H.; Yu, H. Fusion of 3D LIDAR and Camera Data for Object Detection in Autonomous Vehicle Applications. *IEEE Sens J* **2020**, *20*, 4901–4913, doi:10.1109/JSEN.2020.2966034.
14. Liu, C.; Huang, Y.; Rong, Y.; Li, G.; Meng, J.; Xie, Y.; Zhang, X. A Novel Extrinsic Calibration Method of Mobile Manipulator Camera and 2D-LiDAR via Arbitrary Trihedron-Based Reconstruction. *IEEE Sens J* **2021**, *21*, 24672–24682, doi:10.1109/JSEN.2021.3111196.
15. Liu, C.; Huang, Y.; Rong, Y.; Li, G.; Meng, J.; Xie, Y.; Zhang, X. A Novel Extrinsic Calibration Method of Mobile Manipulator Camera and 2D-LiDAR via Arbitrary Trihedron-Based Reconstruction. *IEEE Sens J* **2021**, *21*, 24672–24682, doi:10.1109/JSEN.2021.3111196.
16. Fu, B.; Wang, Y.; Ding, X.; Jiao, Y.; Tang, L.; Xiong, R. LiDAR-Camera Calibration Under Arbitrary Configurations: Observability and Methods. *IEEE Trans Instrum Meas* **2020**, *69*, 3089–3102, doi:10.1109/TIM.2019.2931526.
17. Cui, M.; Zhu, Y.; Liu, Y.; Liu, Y.; Chen, G.; Huang, K. Dense Depth-Map Estimation Based on Fusion of Event Camera and Sparse LiDAR. *IEEE Trans Instrum Meas* **2022**, *71*, 1–11, doi:10.1109/TIM.2022.3144229.
18. Yuan, C.; Liu, X.; Hong, X.; Zhang, F. Pixel-Level Extrinsic Self Calibration of High Resolution LiDAR and Camera in Targetless Environments. *IEEE Robot Autom Lett* **2021**, *6*, 7517–7524, doi:10.1109/LRA.2021.3098923.
19. Csontho, M.; Rovid, A.; Szalay, Z. Significance of Image Features in Camera-LiDAR Based Object Detection. *IEEE Access* **2022**, *10*, 61034–61045, doi:10.1109/ACCESS.2022.3181137.
20. Li, Y.; Deng, J.; Zhang, Y.; Ji, J.; Li, H.; Zhang, Y. A Close Look at the Integration of LiDAR, Millimeter-Wave Radar, and Camera for Accurate 3D Object Detection and Tracking. *IEEE Robot Autom Lett* **2022**, *7*, 11182–11189, doi:10.1109/LRA.2022.3193465.
21. Sualeh, M.; Kim, G.-W. Visual-LiDAR Based 3D Object Detection and Tracking for Embedded Systems. *IEEE Access* **2020**, *8*, 156285–156298, doi:10.1109/ACCESS.2020.3019187.
22. Thanh, D.N.H.; Hieu, L.M.; Enginoglu, S. An Iterative Mean Filter for Image Denoising. *IEEE Access* **2019**, **7**, 167847–167859, doi:10.1109/ACCESS.2019.2953924.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 7 September 2022 doi:10.20944/preprints202209.0109.v1

23. Liu, H.; Hu, F.; Su, J.; Wei, X.; Qin, R. Comparisons on Kalman-Filter-Based Dynamic State Estimation Algorithms of Power Systems. *IEEE Access* **2020**, *8*, 51035–51043, doi:10.1109/ACCESS.2020.2979735.
24. Pereira, P.T.L.; Paim, G.; Costa, P.U.L. da; Costa, E.A.C. da; de Almeida, S.J.M.; Bampi, S. Architectural Exploration for Energy-Efficient Fixed-Point Kalman Filter VLSI Design. *IEEE Trans Very Large Scale Integr VLSI Syst* **2021**, *29*, 1402–1415, doi:10.1109/TVLSI.2021.3075379.
25. Zhang, H.; Zhou, X.; Wang, Z.; Yan, H. Maneuvering Target Tracking With Event-Based Mixture Kalman Filter in Mobile Sensor Networks. *IEEE Trans Cybern* **2020**, *50*, 4346–4357, doi:10.1109/TCYB.2019.2901515.
26. Onat, A. A Novel and Computationally Efficient Joint Unscented Kalman Filtering Scheme for Parameter Estimation of a Class of Nonlinear Systems. *IEEE Access* **2019**, *7*, 31634–31655, doi:10.1109/ACCESS.2019.2902368.
27. Kalman Filter. In Nonlinear Filters; Wiley, 2022; pp. 49–70.
28. Kalman Filter. In Adaptive Filters; John Wiley & Sons, Inc.: Hoboken, NJ, USA; pp. 104–110.
29. Piso, D.; Bruguera, J.D. Variable Latency Goldschmidt Algorithm Based on a New Rounding Method and a Remainder Estimate. *IEEE Transactions on Computers* **2011**, *60*, 1535–1546, doi:10.1109/TC.2010.269.
30. Green, O. Efficient Scalable Median Filtering Using Histogram-Based Operations. *IEEE Transactions on Image Processing* **2018**, *27*, 2217–2228, doi:10.1109/TIP.2017.2781375.