Research on SLAM of indoor mobile robot assisted by AR code landmark

Cong Gu a, Hongjian Zhao b, Junfeng Yuan c, Bowen Teng d, Chenghua Tian e
Beijing Institute of Machinery Industry Automation Beijing, China

a gucong@riamb.ac.cn, b zhaohj95@163.com, c yjf99_1999@163.com, d 13241808527@163.com, e 369585323@qq.com

Abstract. When mobile robots use odometer method to locate and map simultaneously in large-scale indoor scenes, there are some problems, such as large accumulated error of odometer, low positioning accuracy due to unreliable information, and large difference between the mapping results and the real environment. In view of the above problems, this paper proposes a method to correct the accumulated error of odometer with the aid of AR code artificial landmark, so as to meet the requirements of high-precision positioning and mapping of mobile robots. Using the particle filter algorithm based on Rao-Blackwellized to perform SLAM in indoor corridors, the comparison of experimental results proves that the method based on AR code landmarks can obtain high-precision positioning when the robot is running at a long distance, and the mapping results are similar to the real environment, the trajectory accuracy of the mobile robot is improved by about 8%.

Keywords: Mobile robot, Odometer method, SLAM, AR code landmark, RBPF.

1. Introduction
SLAM is the abbreviation of simultaneous localization and mapping, it refers to the subject carrying specific sensors, moving in the environment without prior information, using the sensor to sense the surrounding information, so as to build the environment map, and finally realize its own positioning and navigation [1-3]. It can be seen that sensors are the core factors that affect the robot's perception of environmental information, and accurate positioning and map building are the basis and premise for mobile robot to achieve all operations. Therefore, it is very necessary to use appropriate sensors to perceive the environment information more accurately and improve the accuracy of SLAM. Sensors can be roughly divided into internal sensors and external sensors. Internal sensors include inertial measurement units, coded odometers, etc. External sensors include GPS, camera, sonar, laser rangefinder, etc[4].

Odometer is one of the important internal sensors for pose estimation of autonomous mobile robot, it uses the encoder to obtain odometer data [5]. GRISETTI G[6]and others proposed an improved particle filter method based on Rao-Blackwellized, which used odometer information as a motion model and combined with observation information for trajectory prediction of mobile robots, by adaptive resampling, the number of resampling is reduced, which reduces the risk of particle exhaustion. However, the accuracy of the odometer is easily affected by the environment, which inevitably produces
some errors, such as wheel slip, side slip, idling and other situations in the process of movement. The accumulated errors directly affect the effect of positioning and map construction [7].

The cumulative error of the odometer increases with the increase of the robot's moving range. In addition, in the case of high environment similarity and few scene feature points, the point cloud data features obtained by lidar will also be scarce, which makes the system easy to misjudge the robot's movement as the drift of coded odometer, and the algorithm will reposition the robot, which will lead to positioning errors and inaccurate mapping [8]. In order to achieve high-precision and low computational complexity of positioning and mapping, researchers try to use the method of artificial feature construction to solve the problem [9]. Therefore, the method of landmark assisted positioning is widely used in the construction of large-scale environmental grid maps. Huang Lu[10] designed artificial road signs and established the standard database of road signs, they used the road signs to determine the position of the robot in the environment to correct the error of the odometer. However, the method has some shortcomings such as difficulty in design of road signs, complex establishment of road signs database, and a large amount of landmark data, and the system has large calculation and low recognition efficiency, which cannot meet the actual needs. Hu Zhang fang, Zeng Lin quan and others[11] proposed an adaptive Monte Carlo algorithm incorporating two-dimensional code information. The robot used the absolute position information of two-dimensional code to correct the odometer deviation, and then sampling, which could effectively improve the accuracy of positioning. However, this algorithm has certain requirements for the layout of the two-dimensional code, and the path of the robot is limited by the layout of the two-dimensional code, so the flexibility of the system is poor.

The above method uses the absolute position information of road signs to assist robot positioning, which solves the problem of cumulative error of odometer to a certain extent, but there are still many problems, the complex road signs lead to low recognition efficiency of the system. The two-dimensional code road signs are placed on the ground and other places which are easy to wear, and more two-dimensional codes are needed to ensure the positioning accuracy. The robustness of the algorithm is not high. In view of this, this paper proposes a method to correct the accumulated error of odometer by using the absolute position information of AR code (Augment Reality tag), so as to improve the precise positioning and accurate mapping of mobile robot. In this method, AR code is arranged on the wall or object surface, and the absolute position information of AR code is obtained by using the camera carried by the robot to identify and calculate the relative position of the robot, and then compare with the information of the odometer to determine the error size of the odometer, so as to realize the error correction of the odometer. Through the experiment of actual scene, the proposed method is proved to have good effect.

2. Rao-Blackwellized Particle Filter SLAM Algorithm

Particle filter theory [12-15] originated from Monte Carlo (Monte Carlo) idea, that is, using particle set to represent probability density, and using non-parametric Monte Carlo simulation method to realize recursive Bayesian filtering, which can solve nonlinear system problems that can be described by state space model. Therefore, it is widely used in the research fields of mobile robot localization. In order to solve the computational complexity and low efficiency of ordinary particles in the high-dimensional state estimation problem of SLAM, GRISETTI G[6] and others proposed an improved Particle Filter (RBPF) algorithm based on Rao-Blackwellized. The core of the algorithm is to decompose the estimation of the robot's path attitude and the update of the map. The SLAM problem is decomposed into the product of the posterior probability of the trajectory and the posterior probability of the map, so it can be transformed into the known pose information to solve the problem of updating the map. Formula (1) is the factorization of the joint probability density function by RBPF algorithm.

\[
p(s_t, m/\mathbf{z}_t, u_{t-1}) = p(s_t \mid \mathbf{z}_t, u_{t-1}) p(m/\mathbf{z}_t, s_t) = p(s_t \mid \mathbf{z}_t, u_{t-1}) \prod_{i=1}^{n} p(m_i/\mathbf{s}_t, \mathbf{z}_t, u_{t-1})
\]
Where: $s_t$ represents the robot trajectory at time $t$; $m$ is the observed map; $z_t$ is the observation information at time $t$; $u_{t-1}$ is the control variable from the beginning to time $t-1$, usually the odometer information.

Therefore, RBPF can first use the particle filter method to estimate the posterior probability density function $p(s_t/z_t, u_{t-1})$ of the robot trajectory, and then use the known trajectory and pose to update the probability density function $p(m/z_t, s_t)$. The algorithm flow usually includes the following five parts:

- **State initialization**
  According to the prior probability information provided by the motion model, $N$ particles are randomly selected as the initial state. At this time, the initial weight of each particle is $\omega_0 = 1/N$, and the following initial state set is obtained:

  \[
  \begin{align*}
  \mathbf{x}_0^i & = p(x_0) \\
  \mathbf{x}_t & = \sum_{i=1}^{N} \omega_i^t \mathbf{x}_t^i \\
  \mathbf{p}_t & = \sum_{i=1}^{N} \omega_i^t (p_t^i - (\mathbf{x}_t^i - \mathbf{\bar{x}}_t)) (\mathbf{x}_t^i - \mathbf{\bar{x}}_t)^T
  \end{align*}
  \] (2)

  Where: $i = \{1, 2, \ldots, N\}$, $p(x_0)$ is the initial prior probability.

- **Importance sampling**
  The initial sampling is performed according to the particle distribution rules at the previous moment, and a new particle set in the next moment state is regenerated from the current particle set. Generally, the generation of the next-generation particle set mainly relies on the acquired odometer data information. Therefore, the probability density function of the motion model of the robot can be expressed as:

  \[
  q = p(s_t|u_t, s_{t-1})
  \] (3)

- **Particle weight calculation**
  Combining the above, the weight iteration and recurrence formula of particles can be obtained as

  \[
  \begin{align*}
  \omega_t & = \omega_{t-1} \frac{p(z_t|s_t, s_{t-1})p(s_t|s_{t-1})}{p(z_{t-1}|s_{t-1})p(s_t|s_{t-1})} \\
  & \propto \omega_{t-1} p(z_t|s_t, s_{t-1})
  \end{align*}
  \] (4)

  Where: $z_t, z_{t-1}$ are the observed values at time $t$ and time $t-1$ respectively, $s_t, s_{t-1}$ represent the position and attitude of the robot at time $t$ and time $t-1$ respectively.

- **Resampling**
  In theory, there is a problem of particle degradation in the particle filter theory. This is because in the process of continuous iteration, the sampling result is only determined by a few particles with a larger weight, which leads to the estimation deviation of probability density function is larger and larger. In order to solve the problem of "polarization" of particle weight, resampling is needed. Set up a variable Neff for whether the particle needs to be updated. When the Neff value is lower than the set threshold value, the system will resample and update the particle set to ensure the dispersion of particle weight, so as to ensure the effectiveness of sampling results.

- **Estimation of state parameters**
  The data information obtained from the odometer is used as the motion model to estimate the current pose. Using the data information obtained from the lidar, the two parameters $\mu$ and $\sigma$ of Gaussian estimation are as follows:

  \[
  \begin{align*}
  \mu_t^i & = \frac{1}{\eta} \sum_{j=1}^{k} x_j \cdot p \left( \frac{z_t}{m_{t-1}^i}, x_j \right) p \left( \frac{x_j}{x_{t-1}^i}, u_{t-1} \right) \\
  \sigma_t^i & = \frac{1}{\eta} \sum_{j=1}^{k} p \left( \frac{z_t}{m_{t-1}^i}, x_j \right) p \left( \frac{x_j}{x_{t-1}^i}, u_{t-1} \right) \cdot (x_j - \mu_t^i)(x_j - \mu_t^i)^T
  \end{align*}
  \] (5)
Among them, $\eta^j$ is the normalization parameter, which is specifically expressed as:

$$\eta^j = \sum_{j=1}^{k} p \left( \frac{x_t}{m_{t-1}^j - x_j^p}, x_j^p \right) p \left( \frac{x_j^p}{n_{t-1}^j}, u_{t-1} \right)$$  \hspace{1cm} (6)

It is assumed that the distance between the mobile robot and the two-dimensional code artificial signage is $\gamma$, and the angle is $\alpha$, its own pose is $St$, and the two-dimensional code signage observed by the camera is $\theta mt$. Thus, the observation equation is shown in formula (7), and its Jacobian matrix is shown in formula (8):

$$g(s_t, \theta mt) = \begin{bmatrix} r(s_t, \theta mt) \\ \alpha(s_t, \theta mt) \end{bmatrix} = \begin{bmatrix} \sqrt{(\theta_{mt,x} - s_{t,x})^2 + (\theta_{mt,y} - s_{t,y})^2} \\ \tan^{-1} \left( \frac{\theta_{mt,y} - s_{t,y}}{\theta_{mt,x} - s_{t,x}} \right) - s_{t,\theta} \end{bmatrix}$$  \hspace{1cm} (7)

$$G_{\theta mt} = \begin{bmatrix} \frac{\theta_{mt,x} - s_{t,x}}{q} & \frac{\theta_{mt,y} - s_{t,y}}{q} \\ \frac{\theta_{mt,y} - s_{t,y}}{q} & \frac{\theta_{mt,x} - s_{t,x}}{q} \end{bmatrix}$$  \hspace{1cm} (8)

Where: $q = \left( \theta_{mt,y} - s_{t,y} \right)^2 + \left( \theta_{mt,x} - s_{t,x} \right)^2$  \hspace{1cm} (9)

3. Lidar observation model assisted by AR code landmarks

Nowadays, the method of road sign assisted coding odometer positioning based on two-dimensional code is very popular. Two-dimensional code labels have the advantages of convenient identification, simple information acquisition, and low cost[4]. As one of the typical label two-dimensional codes with high recognition rate, AR code has the characteristics of obvious features, strong error correction ability, easy to make and durable. It is widely used in the mark recognition problem of spatial vision positioning system.

3.1. AR code

AR code is a kind of linear block code, which uses black background and black frame, so it can be distinguished from the surrounding environment, which is more conducive to camera recognition and information extraction. In the middle, the white rectangle representing 1 and the black rectangle representing 0 are distributed according to certain coding rules, thus forming different coding values. By introducing two-dimensional binary bar code mode to correct the confusion between labels, the binary bar code can correct bit errors in detection. Each line includes check bit and data bit, the first, third and fifth are check bit, the second and fourth are data bit. AR code is a special hamming code, the special is that its first reverse, surrounded by easy to identify the black border. The following Fig.1 shows the AR code encoding, the right side shows the binary number represented by each line. The figure below shows the schematic diagram of binary 0010011100 and decimal ID=156[16-17].

![Fig.1 Schematic diagram of AR code coding](image)

3.2. AR code relative pose calculation algorithm

In the process of experiment, this system mainly involves world coordinate system, camera coordinate system and AR code landmark coordinate system. The robot moves in the environment, so we need to choose a reference coordinate system to describe the position and attitude of the robot relative to the
environment, Because the camera is installed on the robot, the position and direction of the camera relative to the environment can also be obtained at the same time. This reference coordinate system is called the world coordinate system.

Using the position information of AR code to obtain the accurate pose information of robot is the core of this paper. The relative position relationship is shown in Fig. 2. At the initial time, the robot is at the initial origin position O1, which is the origin position of the world coordinate system. When the robot moves forward and reaches the O2 position at a certain time, the robot can recognize the AR code and obtain the AR code position information. Since the size of the two-dimensional code is known, the pose of the two-dimensional code relative to the camera can be calculated according to the change of the image obtained by the camera, and the spatial position of the two-dimensional code relative to the camera can also be calculated.

![Fig.2 Schematic diagram of AR code assisted positioning](image)

4. Integration of odometer and AR code landmark positioning

4.1. Single-line lidar SLAM based on odometer

Single-line lidar senses the environment by transmitting laser signal and receiving the signal reflected from the target, after proper processing of the returned signal, it can obtain the relevant information of the target environment, that is, the data obtained from continuous scanning can be used to construct a two-dimensional map. However, when the robot walks in a long and straight corridor with walls on both sides, due to the high similarity of the walls on both sides, the sparse feature points and the high
environmental similarity, it is easy to mistakenly judge the robot's linear motion as encoder odometer drift.

When the mobile robot uses the odometer method to build the map, it mainly depends on the distance between two different times recorded by the photoelectric encoder, so as to calculate the position change. The calculation formula is as follows:

\[
P_n = P_{n-1} + R \begin{bmatrix} \pm D_i \\ 0 \\ 0 \end{bmatrix}
\]

\[
R = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(\phi) & \sin(\phi) \\
0 & -\sin(\phi) & \cos(\phi)
\end{bmatrix}
\times \begin{bmatrix}
\cos(\epsilon) & \sin(\epsilon) & 0 \\
-\sin(\epsilon) & \cos(\epsilon) & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Where: \(P_n, P_{n-1}\) are the position coordinates of the robot at two different times, \(D_i\) is the distance between two positioning, \(\phi, \theta, \epsilon\) are the three attitude angles of the robot, and \(R\) is the external parameter matrix.

Due to the poor relocation ability of laser slam, it is difficult to return to the working state after tracking lost. As a result, the length of the long straight corridor map established is shorter than the real distance, and there will be overlap. In actual applications, this will lead to the distortion and inaccuracy of the built map. Such a map lacking environmental information will cause serious consequences for the robot's own positioning, navigation and path planning.

4.2. AR code landmark assisted SLAM data fusion

The AR code can be used to store information. In the experiment, the absolute position coordinates of the center of labels with different IDs are set, and the laser rangefinder is used to accurately arrange them in the experimental environment. Combined with the absolute position coordinate information of the AR code, if the camera does not scan the AR code during the movement of the mobile robot, the RBPF algorithm based on the odometer method will be used to speculate the pose and build the map. The robot will continue to move forward. When the camera recognizes the landmark, the position information stored in the AR code can be obtained. Combined with the ar_track_alvar function package under the ROS platform [18]. The position relationship between the camera and the AR code in the AR code coordinate system can be calculated, and then the relative position coordinate value between the robot body and the AR code can be obtained, including the translation vector and the rotation quaternion. At this time, the absolute position coordinate value of the AR code and the obtained coordinate value in the relative position relationship are compared, and then the theoretical position of the robot is obtained. Compare this theoretical position with the position fed back by the odometer, if there is an error, it means that the odometer has accumulated error, then the theoretical coordinate value at this time will be assigned to the odometer, and the odometer will start positioning and map construction based on the theoretical value at the next moment. The camera continues to scan and sample until the next two-dimensional code is detected. By repeating the above work, the accumulated error generated by the odometer can be continuously corrected, so that the robot can have more accurate position coordinates.

5. Experimental verification and analysis

5.1. Experimental platform

In order to verify the effectiveness of the algorithm proposed in this paper, the omnidirectional McNum wheeled mobile robot as shown in Fig. 2 below is used, the bottom layer is equipped with Intel dual-core processor development board, the STM32 MCU is used as the driver board, and the flat panel display screen is connected to display the mapping results. The upper layer is equipped with RPLIDAR-A1 2D lidar and 1080P-60HZ camera. The lidar can scan the surrounding environment 360 degrees, and the maximum measurement range is around 8m. The camera is installed obliquely 45 degrees above the front of the car, and the height from the ground is 16cm, so as to ensure that it can accurately identify the AR code. Except for the camera installed on the top, other hardware devices have been packaged, as
shown in Fig. 3 below. In order to facilitate identification, the AR code is made with a side length of 0.15m, and it is attached to a wall as high as the camera, and deployed at a distance of 10 meters, as shown in Fig. 4 below.

Fig.3 Experimental platform

Fig.4 Schematic diagram of AR code pasting

5.2. Results and analysis

This paper selects a 23-meter-long straight corridor and a 140-meter square corridor for experiments. The environment of the experimental corridor is shown in Fig. 5 and Fig. 8. Take the blue dot in the figure as the starting point of the robot, which is the origin of the world coordinate system, and the world coordinate system is established. The forward direction is the X axis, the right direction is the Y axis, and the upward direction is the Z axis. In the long and straight corridor in Figure 5, 2 AR code labels are set up according to the requirement of one per ten meters. In the square corridor in Fig.8, a total of 14 tags are arranged, all of which are arranged on the wall.

The robot walks in the long and straight corridor in Figure 5 and performs SLAM experiments. The corridor is 23 meters long and 1.9 meters wide. The two AR codes are ID0 and ID1. The position coordinates of ID0 are (10.00, 0.80, 0.23), and the position coordinates of ID1 are (20.00, 0.80, 0.23).

Fig.5 Experimental environment 1

Use the Gmapping package in the ROS system to build a raster map. Record the results of the robot’s map creation with or without AR code landmarks, and compare the two established raster maps with the real map, as shown in the red circle in Fig. 6 below. It can be found that when AR code landmarks are not set, the raster map has the problem that the vertical length of the map is smaller than the size of the
real map due to the inclination of the map, and there is also a problem that some of the rasters are blurred. In the case of setting artificial road signs, as shown in Fig. 7 below, the map created by the mobile robot is relatively clear and there is no inclination, and the mapping results are basically close to the real environment.

![Fig.6 Map results without landmarks](image1)

![Fig.7 Map results with landmarks](image2)

In order to verify the feasibility of the algorithm effectively, the experiment is carried out in the real scene in Figure 8. The square corridor is 140 meters long and 1.8m wide. A total of 14 AR codes are set, ID0 to ID13, and corresponding location coordinate information is added.

![Fig.8 Experimental environment 2](image3)

Fig. 9 records the mapping results of the robot without AR code landmarks. It can be seen that there is a skew distortion in the construction of the map, and there is an offset in the closed-loop process. Some of the results are quite different from the actual situation.
On the contrary, the result of mapping with AR code landmarks is shown in Fig. 10. Compared with Fig. 9, some of the locations are quite different. With landmarks, the positioning and mapping are more accurate. The corridor in the figure is similar to the real environment 98%, and the accuracy of the error between the robot's trajectory length and the actual length during the mapping process is increased by approximately 8%.

6. Conclusion
In this paper, a method based on AR code is proposed to assist robot precise positioning and mapping. Experiments are carried out in two different environments. The mobile robot calculates its relative position information relative to the two-dimensional code by recognizing the landmark of AR code, and compares it with the absolute position information provided by AR code, so as to obtain the absolute position information of the mobile robot, and then compares it with the odometer data, so as to correct the error of the odometer. Experiments show that this method can reduce the cumulative error generated by the odometer in the process of navigation and map creation, and correct the error in time, and effectively improve the accuracy of indoor map creation. In addition, this paper arranges the two-dimensional code design on the wall, which is not only flexible, but also not easy to wear and low maintenance costs. This method combines visual landmark information and odometer data information. The accuracy of robot positioning needs to be further improved. The use of multi-sensor fusion for autonomous positioning, navigation and mapping of mobile robots is the main method to improve its accuracy in the future. Therefore, this paper will further study multi-sensor data information fusion on the basis of existing algorithms, and provide the robot with more and more accurate prior information, and then improve the robot's positioning accuracy.

Acknowledgments
Technical development fund project of Beijing Institute of Machinery Industry Automation.
References

[1] Gao Xiang, Zhang Tao. Visual SLAM: From Theory to Practice [M]. Beijing: Publishing House of Electronics Industry, 2017.
[2] Ren Mingyu, Chen Wanmi, Zhang Yuanyuan. Research on a Robotic SLAM Method Fusing Laser and Vision Information [J]. Electronic Measurement Technology, 2019, 42(13): 92-97.
[3] GRISETTI G, STACHNIS C, BURGARD W. Improving Grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling [C]. IEEE International Conference on Robotics & Automation. IEEE, 2005: 2432-2437.
[4] Zhang Anfeng, Wu Tao, Yang Congcong. Research on Aided Localization of Mobile Robot Autonomous Navigation Matrix QR Code [J]. Software, 2019, 40(06): 180-184.
[5] Ran Teng, Yuan Liang, Wu Jingjiang, Luo Gao. Indoor Mobile Robot SLAM Assisted by Artificial Roadpost [J]. Machinery Design & Manufacture, 2020(07): 246-249.
[6] GRISETTI G, STACHNIS C, BURGARD W. Improved techniques for grid mapping with Rao-Blackwellized particle filters [J]. IEEE Transactions on Robotics, 2007, 23(1): 34-46.
[7] Luan Jianing, Zhang Wei, Sun Wei, Zhang Ao, Han Dong. High Precision Localization Algorithm Based on QR Code Vision and LiDAR Fusion [J/OL]. Computer Application, 2021.
[8] Yonghoon Ji and Atsushi Yamashita and Hajime Asama. RGB-D SLAM using vanishing point and door plate information in corridor environment [J]. Intelligent Service Robotics, 2015, 8(2): 105-114.
[9] LEE S, TEWOLDE G, LIM J, et al. QR-code based Localization for Indoor Mobile Robot with validation using a 3D optical tracking instrument [C]/Proceedings of the 2015 IEEE International Conference on Advanced Intelligent Mechatronics. Piscataway: IEEE, 2015: 965-970.
[10] Huang Lu, Zhu Ming. Navigation Method of Mobile Robot Assisted by Monocular Vision Artificial Signpost [J]. Application of Computer Systems, 2018, 27(01): 106-112.
[11] Hu Zhenfang, Zeng Linquan, Luo Yuan, Luo Xin, Zhao Liming. Adaptive Monte Carlo Location Algorithm Integrating QR Code Information [J]. Computer Applications, 2019, 39(04): 989-993.
[12] DELLAERT F, DIETER D, BURGARD W, et al. Monte Carlo localization for mobile robots [C]/Proceedings of the 1999 IEEE International Conference on Robotics and Automation. Piscataway: IEEE, 1999: 1322-1328.
[13] Thrun S, Fox D, Burgard W. Robust monte carlo localization for mobile robots [J]. Artificial Intelligence, 2001, 128(1): 99-141.
[14] KONG Hongshan, LI Xiaopeng, YU Bin. An Improved Algorithm of SIR Particle Filtering [J]. Computer Engineering and Design, 2020, 41(07): 1899-1904.
[15] Zhou Xu. Research on SLAM Algorithm Based on Improved Particle Filtering [D]. Nanjing University of Science and Technology, 2014.
[16] Tao Sha Lv, Wen Tang, Tao Ruan Wan, Yaolin Zhu, Tong Wu. Research on remote target recognition method in augmented reality [J]. Software Engineering, 2015, 2015(04): 7-8.
[17] Yang Meng, He Yaxuan, Yan Deli, Zhang Yong, Wang Weiming. Cylindrical AR Code Recognition Algorithm Based on Arc Modified Filtering [J]. Journal of University of Science and Technology of China, 2018, 48(01): 1-6.
[18] Hu Chunxu. ROS Robot Development Practice [M]. Beijing: China Machine Press, 2018: 224-228.