Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 9841018, 1 page
https://doi.org/10.1155/2023/9841018

Retraction

Retracted: FastQR: Fast Pose Estimation of Objects Based on Multiple QR Codes and Monocular Vision in Mobile Embedded Devices

Wireless Communications and Mobile Computing

Received 28 November 2023; Accepted 28 November 2023; Published 29 November 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Y. Yan, Y. Liang, Z. Zhou, B. Jiang, and J. Xiao, "FastQR: Fast Pose Estimation of Objects Based on Multiple QR Codes and Monocular Vision in Mobile Embedded Devices," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 9481190, 9 pages, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9481190, 9 pages
https://doi.org/10.1155/2021/9481190

Research Article

FastQR: Fast Pose Estimation of Objects Based on Multiple QR Codes and Monocular Vision in Mobile Embedded Devices

Yuheng Yan, Yiqiu Liang, Zihan Zhou, Bin Jiang, and Jian Xiao

Nanjing University of Posts and Telecommunications, Nanjing, China

Correspondence should be addressed to Jian Xiao; xiaoj@njupt.edu.cn

Received 8 July 2021; Accepted 9 September 2021; Published 30 September 2021

Academic Editor: Balakrishnan Nagaraj

In recent years, the pose estimation of objects has become a research hotspot. This technique can effectively estimate the pose changes of objects in space and is widely used in many mobile devices, such as AR/VR. At present, mainstream technologies can achieve high-precision pose estimation, but the problem of that of multiple irregular objects in mobile and embedded devices under limited resource conditions is still challenging. In this paper, we propose a FastQR algorithm that can estimate the pose of multiple irregular objects on Renesas by utilizing homography method to solve the transformation matrix of a single QR code and then establish the spatial constraint relationship between multiple QR codes to estimate the posture of irregular objects. Our algorithm obtained a competitive result in simulation and verification on the RZ/A2M development board of Renesas. Moreover, the verification results show that our method can estimate the spatial pose of the multiobject accurately and robustly in distributed embedded devices. The average frame rate calculated on the RZ/A2M can reach 28 fps, which is at least 37 times faster than that of other pose estimation methods.

1. Introduction

With the development of AR/VR and control technology, object pose estimation is needed in manifold scenes. A large number of low-cost small embedded platforms also need to carry out object spatial pose estimation, such as in the exhibition, education, and other fields [1, 2]. At present, most of the estimation of object pose depends on the binocular camera [3] or RGB-D camera [4]. This type of camera is large and requires a computing platform with a certain amount of computing power, which is not conducive to the installation on small devices. There are also inertial sensors on objects to complete pose detection [5] or the comprehensive use of IMU, binocular camera, and RGB-D camera [6], but this method has certain difficulties in data fusion and filtering. Moreover, using convolution neural network to estimate the position and pose of objects [7] has also become a research focus, which brings accuracy, but requires high computational power of the platform, and even needs to be accelerated by GPU.

In many scenarios, like AR, it is necessary to realize the fast pose estimation of objects on small portable devices, so the pose estimation of objects needs a convenient, fast, and low-cost solution. With the development and popularization of QR code technology [8] in recent years, QR code has been widely used in various aspects, such as QR code payment and QR code business cards. Most of these applications use the function of QR codes carrying text information, but at the same time, QR codes can also provide accurate pose information. At present, in the existing research, it has been possible to use QR codes as visual markers, utilizing the position, number, and other information stored in the QR code to complete the navigation [9, 10] or using the calculated position and attitude information to estimate the position and pose of the camera [11].

This paper proposes a FastQR algorithm that can estimate the pose of multiple irregular objects on Renesas by utilizing QR codes and a monocular camera and then estimate the posture of irregular objects. This method only requires low resolution camera, which reduces the amount

of computation with the characteristics of low cost, high accuracy, good robustness, low computing power requirements, fast speed, and so on, and making the multiobject pose estimation has a wider range of application scenarios. The results also verify the reliability of our method. At the same time, the highest frame rate calculated on the Renesas RZ/A2M development board can reach 45 fps, and the average frame rate can reach 28 fps.

The contributions of this paper are as follows:

(1) It is proposed to use QR codes to realize a high frame rate and high accuracy pose estimation on equipment with limited resources

(2) It is possible to estimate the pose of multiple objects simultaneously under the premise of ensuring the frame rate and accuracy

(3) The proposed algorithm is tested on the simulation platform and the real object, and the result is competitive

2. The Principle of Position and Pose Estimation of Objects

2.1. Identification of QR Codes. The recognition and decoding algorithm of the QR code used in our system is AprilTag [12]. To further improve the recognition speed, the Dynamically Reconfigurable Processor (DRP) [13] on RZ/A2M is used to further accelerate the image processing steps, including adaptive binarization, Gaussian blur, and edge detection in the process of QR code recognition.

2.2. Solution of Position and Pose of QR Codes. Using the AprilTag recognition algorithm, the coordinates of the QR code in the image coordinate system can be found out. Meanwhile, the transformation matrix $H = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix}$ [14] from space coordinates of the QR code to plane coordinates can be obtained.

The projection matrix of the camera can be expressed as $P = K[R \mid T]$, which means point X in the space can be transformed into a pixel on the image through the matrix. The rotation matrix is $R = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix}$. The translational vector is $T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$. The camera internal parameter matrix is $K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$.

P is the projection matrix. In camera internal parameter matrix K, f_x and f_y represent the focal length of the camera in the x and y directions. c_x and c_y represent the position of the intersection of the camera's principal axis on the image plane. At the same time, it is set that the plane where the QR code is located is on plane $X - Y$ $(Z = 0)$, of which the center is the origin of the coordinate. Hence, the following is concluded as:

$$x = K[R \mid T] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}. \tag{1}$$

To simplify the calculation, removing the third column of R in formula (1) can have

$$x = K[r_0 \quad r_1 \quad T] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \tag{2}$$

r_0 and r_1 in formula (2) is the first and second columns of R.

The homography matrix H from the point on the space plane to the point on the image is constructed.

Suppose

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix}, \tag{3}$$

where λ represents the scaling ratio, of which $[X' \quad Y' \quad 1]^T$ is the corner of the zoom unit QR code; the following can be concluded:

$$x = K[r_0 \quad r_1 \quad T] \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \tag{4}$$

It is known that the internal parameter matrix of the camera is K and the single transformation matrix is H. Let $K[\lambda r_0 \quad \lambda r_1 \quad T] = K[r_0' \quad r_1' \quad T]$, the equation can be concluded as the following:

$$\begin{aligned} f_x r_{00}' + c_x r_{20}' &= h_{00}, \\ f_x r_{01}' + c_x r_{21}' &= h_{01}, \\ f_x t_x + c_x t_z &= h_{02}, \\ f_y r_{10}' + c_y r_{20}' &= h_{10}, \\ f_y r_{11}' + c_y r_{21}' &= h_{11}, \\ f_y t_y + c_y t_z &= h_{12}, \\ r_{20}' &= h_{20}, \\ r_{21}' &= h_{21}, \\ t_z &= h_{22}. \end{aligned} \tag{5}$$

$[r_0' \quad r_1' \quad T]$ can be got, and then, the unit is dealt with the following steps:

$$r_0'' = \frac{r_0'}{\sqrt{\|r_0'\|\|r_1'\|}}, r_1'' = \frac{r_1'}{\sqrt{\|r_0'\|\|r_1'\|}}, T'' = \frac{T}{\sqrt{\|r_0'\|\|r_1'\|}}. \tag{6}$$

It is shown that r_0'' and r_1'' after the dealt are the same as r_0 and r_1, while T and T' are different from each other. For the same QR code on the camera image, T represents the translational vector of the actual size of the QR code in the direction of R, and T represents the translational vector of the QR code whose actual size is a square in the same direction. Because of the proportional scaling of the same QR code square in the direction, the translational vector from the camera to the actual QR code can be calculated by the size of the actual one. If the width of the QR code is ω, the translational vector from the camera to the actual QR code is $T = \omega T'$. The translational vector T is also the coordinator of the QR code center in the camera coordinate system.

2.3. Position and Pose Transformation from Multiple QR Codes Coordinate System to Object Coordinate System. By pasting multiple QR codes on an object, the omnidirectional pose estimation of the object can be realized, and the QR code can be avoided being blocked from vision. Meanwhile, according to the position relation of multiple QR codes on the same object, the overall pose information of a single QR code can be obtained from one of a single QR code. The rotation matrix R and translational vector T mentioned in Section 2.2 are the rotation and shift transformation of the QR code coordinate system. In order to map the pose of the QR code to the pose of the object, it is necessary to convert the QR code into the object coordinate base vector group.

As for the pose of the object, it is assumed that the six faces of a hexahedral are affixed with QR codes, respectively, and the center of the QR code coincides with the center of each surface, as shown in Figure 1 below. Suppose the basis vectors of the object coordinate system X, Y, and Z are $\vec{e_x}$, $\vec{e_y}$, $\vec{e_z}$ and only consider the change of the direction of the base vector of the coordinate system.

When the coordinate system of the QR code is consistent with the direction of axis X, Y, and Z of the object coordinate system, as shown in the front of the stereo in Figure 1, suppose the base vectors of the three axes under the object coordinate system are $\vec{e_{x0}}, \vec{e_{y0}}, \vec{e_{z0}}$, the rotation matrix can directly represent the pose of the object, that is

$$\begin{bmatrix} \vec{e_{x0}} \\ \vec{e_{x0}} \\ \vec{e_{x0}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{e_x} \\ \vec{e_x} \\ \vec{e_x} \end{bmatrix}. \tag{7}$$

When the coordinate system of QR code is inconsistent with the direction of axis X, Y, and Z of the object coordinate system, as shown on the right of the stereo in Figure 1, suppose the base vectors of the three axes under the object coordinate system are the direction of the Y-axis of the QR code coordinate system is consistent with the direction of the Y-axis of the object coordinate system; the direction of the X-axis of the QR code coordinate system is opposite to the one of the Z-axis of the object coordinate system, and the direction of the Z-axis of the QR code coordinate system is consistent with the direction of the X-axis of the object coordinate system, that is

$$\begin{bmatrix} \vec{e_{x1}} \\ \vec{e_{x1}} \\ \vec{e_{x1}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \vec{e_x} \\ \vec{e_y} \\ \vec{e_z} \end{bmatrix}. \tag{8}$$

Similarly, when the other surfaces of the QR code are recognized, the pose of the object can be obtained from any angle according to the relationship between the known QR code coordinate system and the object coordinate system.

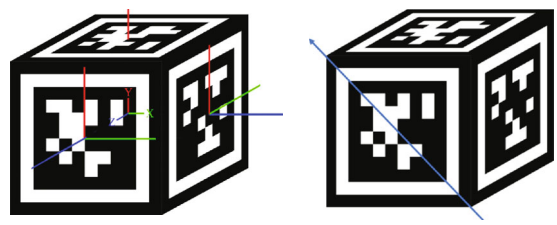
As for the position of the object, suppose the coordinate of a vector in the QR code coordinate system is $(x_0, y_0, z_0)^T$, and the coordinate in the camera coordinate system is $(x_0', y_0', z_0')^T$, the convert relationship can be obtained as follows:

$$\begin{bmatrix} x_0' \\ y_0' \\ z_0' \end{bmatrix} = R \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + T. \tag{9}$$

The convert relationship can be obtained as follows if $(x_0', y_0', z_0')^T$ is transformed into $(x_0, y_0, z_0)^T$:

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = R^{-1} \left(\begin{bmatrix} x_0' \\ y_0' \\ z_0' \end{bmatrix} - T \right). \tag{10}$$

The coordinate of the object center in the camera coordinate system can be obtained by formula (9). For example, in Figure 1, the coordinate of the object center in the QR code coordinate system can be regarded as $(0, 0, -\omega/2)^T$, of which ω stands for the edge length of the pasted QR code. The coordinate of the object center in the camera coordinate system can be acquired by the transformation of formula (11). The transformation from any point in the QR code coordinate system to the camera coordinate system can also be obtained by the transformation of formula (9), so the transformation from the base



(a) (b)

Figure 1: The model of tag cube. (a) Schematic diagram of the QR code and object coordinate system. (b) The model used for testing the pose measurement.

vector set of the QR code coordinate system to the camera coordinate system can be obtained.

$$\begin{bmatrix} x_0' \\ y_0' \\ z_0' \end{bmatrix} = R \begin{bmatrix} 0 \\ 0 \\ -\omega/2 \end{bmatrix} + T. \tag{11}$$

From the analysis above, what is known is that the spatial pose of a three-dimensional object can be estimated through the relative position of the position and pose information of the identified QR code and multiple QR codes on the known object.

2.4. Multiple QR Code Pose Estimation of Irregular Rigid Bodies. Considering that it is difficult to get the transformation relationship from QR code coordinate system to object coordinate system on an irregular rigid body, the system can calculate the transformation relationship among QR codes according to the QR code pasted on the object, and the concrete principle is as follows.

By pasting multiple QR codes on an object, it is necessary to ensure that each QR code can be recognized along with another QR code at a certain angle of view, which is called "connection," and that a QR code can be "connected" to all QR codes according to the above conditions, and the coordinate system of one QR code is used as the object coordinate system. The transformation relationship between the coordinate system base vector group of the QR code connected to the QR code and the base vector group of the QR code coordinate system is calculated by taking one of the QR codes in a coordinate system as the object coordinate system.

Again, taking the cube in Figure 1 as an example, remember that the QR code on the front is A and the QR code on the right, back, left, top, and bottom are B, C, D, E, and F, respectively. Firstly, the camera faces the QR code A on the front of the cube, where the coordinate system is regarded as the object coordinate system to rotate the cube to the left. When the camera rotates to about 45 degrees, the QR codes marked A, and B can be both recognized, and the transformation relationship from QR code A coordinate system to QR code A coordinate system can be got. To simplify the calculation, the pose and position transformations are considered separately, which means only the change of the base vector direction of the coordinate system is considered when the position transformation is carried out, and only the change of the origin of the coordinate system is considered when the position transformation is carried out.

As for the origin of the coordinate system, the coordinate of the origin of QR code A coordinate system under the coordinate system of QR code B is $T_0 - T_1$. For the coordinate system, the following transformation formula can be obtained:

$$\begin{aligned} \vec{e_{x0}} &= R_0 {R_1}^{-1} \vec{e_{x1}}, \\ \vec{e_{y0}} &= R_0 {R_1}^{-1} \vec{e_{y1}}, \\ \vec{e_{y0}} &= R_0 {R_1}^{-1} \vec{e_{z1}}. \end{aligned} \tag{12}$$

R_0 and T_0 represent the rotation matrix and shift vector of QR code A, and $\vec{e_{x0}}, \vec{e_{y0}}, \vec{e_{z0}}$ is the representation of the base vector group of QR code A coordinate system in the camera coordinate system; R_1 and T_1 represent the rotation matrix and shift vector of QR code B, and $\vec{e_{x1}}, \vec{e_{y1}}, \vec{e_{z1}}$ is the representation of the base vector group of QR code B coordinate system in the camera coordinate system.

Keeping rotating the cube until QR code C and QR code B can be both recognized. For the origin of the coordinate system, the coordinate of the origin of the coordinate system of the QR code B under the coordinate system of the QR code C is $T_1' - T_2$. Combined with $T_0 - T_1$, the coordinate of the origin of the coordinate system of QR code 0 under the coordinate system of the QR code B and the coordinate of the origin of the coordinate system of QR code A under the coordinate system of the QR code C can be obtained: $(T_1' - T_2) + (T_0 - T_1)$. According to formula (9), the coordinate of the QR code A coordinate system in the camera coordinate system can be acquired. Similarly, the transformation relationship from QR code C coordinate system to the QR code B coordinate system can be obtained from formula (13).

$$\begin{aligned} \vec{e_{x1}}' &= {R_1}' {R_2}^{-1} \vec{e_{x2}}, \\ \vec{e_{y1}}' &= {R_1}' {R_2}^{-1} \vec{e_{y2}}, \\ \vec{e_{z1}}' &= {R_1}' {R_2}^{-1} \vec{e_{z2}}. \end{aligned} \tag{13}$$

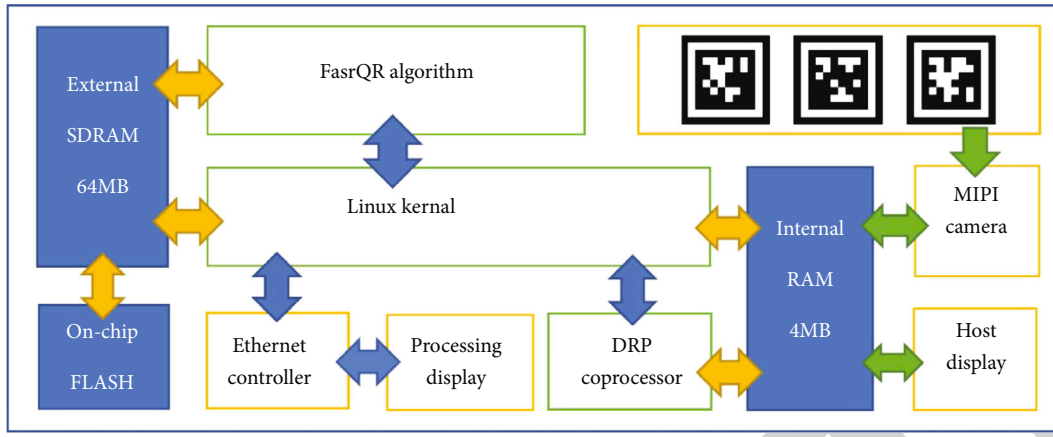


Figure 2: Systematic structure diagram.

R_1' and T_1' are the rotation matrix and shift vector of QR code B, and $\vec{e_{x1}}', \vec{e_{y1}}', \vec{e_{z1}}'$ is the representation of the base vector group of QR code B coordinate system in the camera coordinate system. Combined with formula (12), the transformation relationship from QR code C coordinate system to QR code B coordinate system can be obtained, and the rotation matrix is $R_0 R_1^{-1} R_1' R_2^{-1}$. Continue to rotate the cube, the transformation relationship from all the QR code coordinate systems on the object to QR code A coordinate system can be obtained.

The specific initialization process goes as follows: firstly, enter the number of QR codes posted on the object and the number of objects, then initialize each object one by one. Align any QR code on the first object with the camera, and the system will record the QR code coordinate system as the object coordinate system; rotate the object slowly so that the system can recognize all the QR codes posted on the object, and each new QR code needs to be recognized at the same time as an existed QR code. According to the coordinate system transformation relationship between each new QR code and the existed QR code that has been identified with it as well as the transformation relationship between the coordinate system of the existed QR code and the object coordinate system, the transformation relationship between the new QR code and the object coordinate system is obtained and recorded in the system through the principle of the above transformation relationship, and the initialization of an object is completed.

3. Verification Method and Environment

3.1. Verification Preparation. The system platform adopts Renesas RZ/A2M development board, uses rza_linux-4.19 provided by Renesas with C programming language, which connects to the RGB display through the HDM interface on the board, and sends the pose data to another PC through the network interface on the board to display the pose model. The camera has a Raspberry pi cameraV2 with a resolution of 640 ∗ 480. The QR code adopts Tag36H11 code in AprilTag. Camera calibration and post-data processing use Matlab2019, and the pose system model display uses Processing3. The system structure diagram is shown in Figure 2.

The validation environment is built, as shown in Figure 3 below. The validation environment includes a Raspberry pi camera V2, Renesas RZ/A2M development board, an LCD display, Processing3 display, and the objects pasted with QR codes.

The software flow chart of the system platform is shown in Figure 4. The software mainly includes camera calibration, QR code detection, transformation matrix calculation, and object pose estimation.

3.2. Means of Verification. When checking the position measurement, the QR code is 4 cm ∗ 4 cm, and it is pasted on a regular hexahedral surface of 6 cm∗6 cm. The measurement accuracy is obtained by measuring the position coordinates of the object in the camera coordinate system and comparing it with the position coordinates obtained by the system. It is also necessary to calibrate the position of the camera before testing the attitude measurement, display a QR code in the center of the display, and place the camera in the position of 1.2 m in front of the display. Adjust the angle of the camera until the angle of the measured shaft angle is 0.

In the process of checking the pose measurement, in light of the spatial pose of the object is difficult to measure accurately, Processing3 is used to display a regular hexahedral model with 6 cm the edge length of the model. Different QR codes are displayed on the six faces of the regular hexahedral, and the axial angle of the model pose is brought out while adjusting the model. The accuracy of the measurement is obtained by comparing the pose of both the model and the measurement. The model is shown on Processing3, rotating the model around the arrow direction shown in Figure 1.

4. Results

4.1. Results and Errors of Object Pose Estimation. Considering that the edge length of QR code is 4 cm, the change range of z-axis direction is between 0.1 m and 3.0 m when $x = y = 0.0$ m, and the step size is 0.1 m with 30 groups of data. In light of the angle and view of the camera, when $z = 1.2$

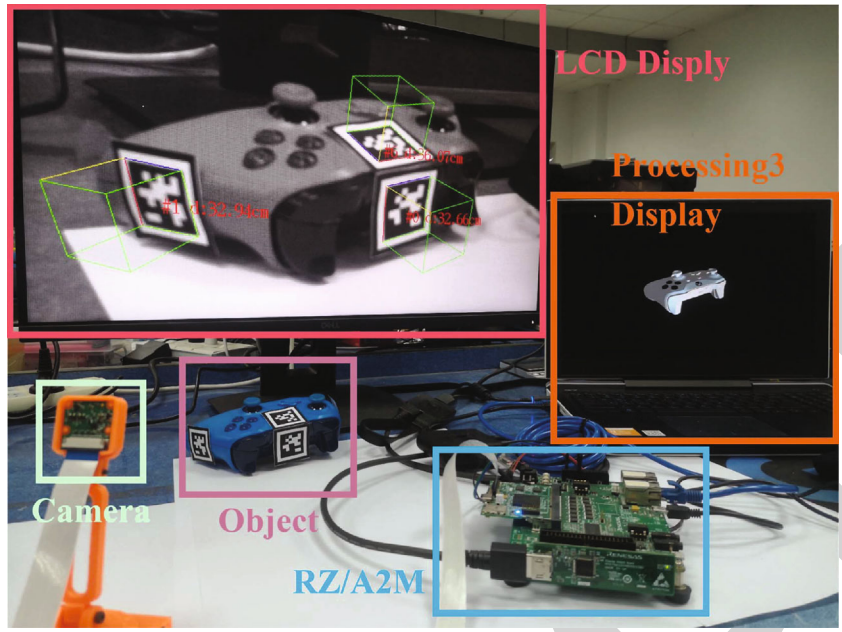


Figure 3: Verification environment.

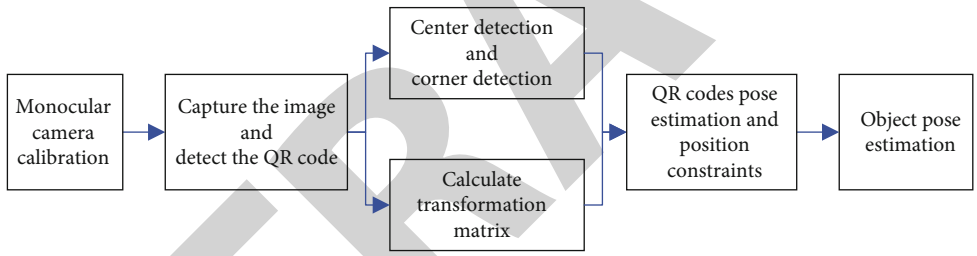


Figure 4: Software flow chart.

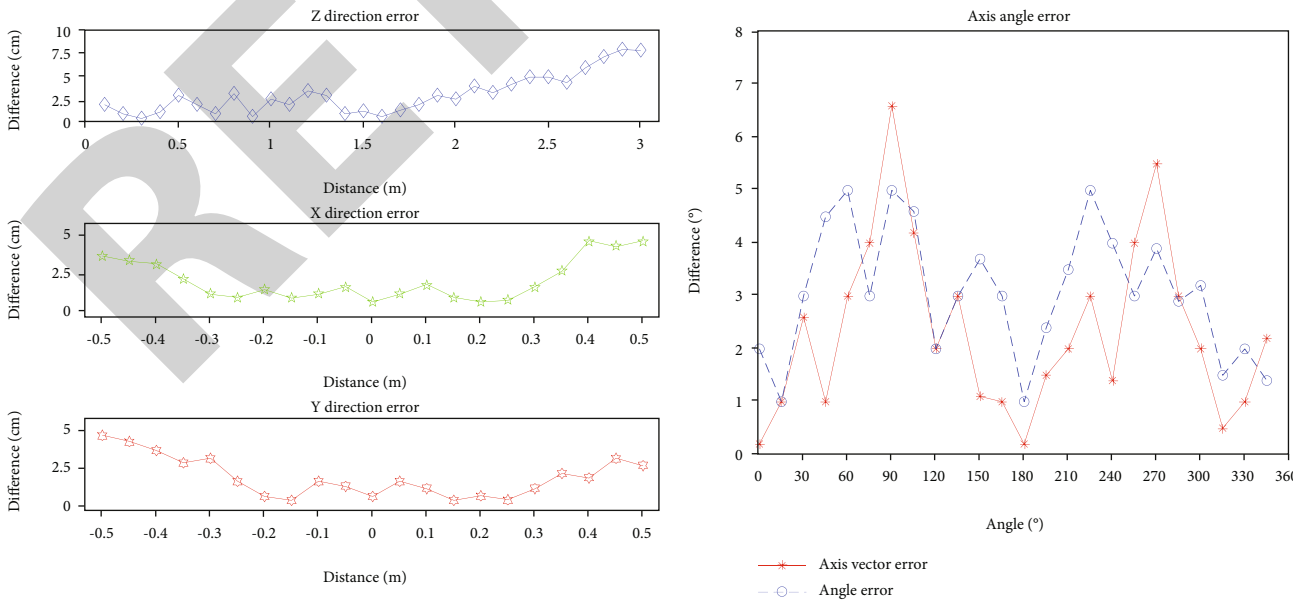


Figure 5: Pose estimation error. (a) The error of position estimation. (b) The error of attitude.

Table 1: The average time spent on object estimation.

Number of QR codes	QR code recognition time (ms)	Pose solution time (ms)	Total calculation time (ms)	Average frame rate (fps)
1	12	10	22	
2	21	15	36	28.57
3	26	21	47	

Table 2: Velocity comparison with other object pose estimation methods.

Number	Pose estimation method	Calculating platform	Average frame rate (fps)
1	FastQR (ours)	Renesas RZ/A2M (with the DRP)	28
2	Estimation of the 3D pose of the object by 3D model of an object [16]	A 2.83 GHz CPU (Q9550) and an NVIDIA GTX 280 GPU	4
3	Estimation of position and pose of the camera by AprilTag [17]	PC	8
4	Pose estimation of objects using fusion visual and inertial data [18]	A 3.60 GHz CPU and NVIDIA Cmadro 4000 GPU	10
5	6DoF pose estimation of objects by geometric information [19]	16-core Intel(R) Xeon(R) E5-2637 CPU and GeForce GTX 1080 GPU	30
6	Seamless single shot 6D object pose prediction [20]	Desktop with a Titan X GPU	50

Figure 6: Effect diagram of the pose estimation of single and multiple irregular objects.

m, the error of the x-axis and y-axis is measured, and the range of variation is between -0.50 m and 0.50 m with 21 groups of data. The error of the angle is calculated by, respectively, calculating the angle between the axis of the real value and the axis of the measured value and by making the difference. The results are shown in Figure 5.

From the results, it can be seen that the error between the real value and the measured one is so small that the error in the z-direction is not more than 9 cm, and the error in the x and y directions is not more than 5 cm. The angle error between the true value of the axis angle and the measured one is not more than 7°, and the angle error is not more than 5°. Due to the limitation of QR code size, camera resolution, and illumination, the accuracy of pose estimation is only more accurate, and the error will increase to a certain extent with the increase of the distance from the object to the camera.

4.2. The Estimated Velocity of Object Position. The system performance is average in terms of accuracy [15], but it can already meet the needs of 6DoF pose estimation for objects in most cases. In terms of the calculation speed, the system has very excellent performance, and its highest main frequency of the Renesas RZ/A2M development board used in this system is 528 MHz, and its captured image resolution is $640 * 480$. The average time of object estimation is shown in Table 1 below. When using DRP on RZ/A2M to accelerate some image processing steps, the fastest time for pose estimation is about 22 ms.

Compared with most of the methods that do not use visual markers to estimate the pose of objects, this method requires less calculation power and does not need to be accelerated by GPU. By using the QR code as the visual marker, the difficulty of finding feature points is avoided, and the resolution of the input image is less required. It can be implemented on MPU or CPU with low main frequency and can still maintain a smooth and available frame rate of about 28 Hz. The velocity comparison with other object pose estimation methods is shown in Table 2 below.

4.3. Position and Pose Estimation of Irregular Objects. Here, taking the XBOX handle as an example, seven QR codes are pasted on the handle to meet the requirements of Section 2.4. Note that the QR code should be pasted smoothly to avoid affecting the recognition effect, and no duplicate QR code should be pasted. The pose estimation effect for single and multiple targets is shown in Figure 6 below.

5. Conclusion

The system uses a monocular camera and multi-QR code to estimate the spatial pose of the object, which has extremely low cost, small volume, low computing power, fast speed, good robustness, and so on. This can meet the basic requirements for pose estimation. Also, at the expense of a certain precision, the calculation power requirement of object spatial pose estimation is lowered, and the calculation speed is greatly improved. It can be realized on many mobile embedded devices' cores, such as MPU or CPU with low main frequency and small memory. Compared with the one using GPU to accelerate pose estimation, the method has a higher energy efficiency ratio and greatly expands the application scene of object pose estimation. This scheme can also be further extended to complex multijoint objects in distributed embedded computing regions.

As future work, we plan to extend this method to achieve fast pose estimation on multijoint targets. By posting QR codes to each joint and connecting rod of multijoint objects, the spatial pose estimation of complex multijoint objects can be realized. We will further study to improve the accuracy of pose estimation without reducing the computing speed.

Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (grant number: 61974073).

References

[1] P. Daponte, L. De Vito, F. Picariello, and M. Riccio, "State of the art and future developments of the Augmented Reality for measurement applications," *Measurement*, vol. 57, pp. 53–70, 2014.

[2] P. Khandelwal, P. Swarnalatha, N. Bisht, and S. Prabu, "Detection of features to track objects and segmentation using GrabCut for application in marker-less augmented reality," *Procedia Computer Science*, vol. 58, pp. 698–705, 2015.

[3] Z. Luo, K. Zhang, Z. Wang, J. Zheng, and Y. Chen, "3D pose estimation of large and complicated workpieces based on binocular stereo vision," *Applied Optics*, vol. 56, no. 24, pp. 6822–6836, 2017.

[4] C. Choi and H. I. Christensen, "3D pose estimation of daily objects using an RGB-D camera," in *Proceedings of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3342–3349, 2012.

[5] M. B. Alatise and G. P. Hancke, "Pose estimation of a mobile robot based on fusion of IMU data and vision data using an extended Kalman filter," *Sensors*, vol. 17, no. 10, p. 2164, 2017.

[6] C. X. Guo and S. I. Roumeliotis, "IMU-RGBD camera 3D pose estimation and extrinsic calibration: observability analysis and consistency improvement," in *Proceedings of 2013 IEEE International Conference on Robotics and Automation*, pp. 2935–2942, Karlsruhe, Germany, 2013.

[7] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," 2017, https://arxiv.org/abs/1711.00199.

[8] S. Tiwari, "An introduction to QR code technology," in *Proceedings of 2016 International Conference on Information Technology (ICIT)*, pp. 39–44, Bhubaneswar, India, 2016.

[9] H. Zhang, C. Zhang, W. Yang, and C. Chen, "Localization and navigation using QR code for mobile robot in indoor environment," in *Proceedings of 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2501–2506, Zhuhai, China, 2015.

[10] G. Zhenglong, F. Qiang, and Q. Quan, "Pose estimation for multicopters based on monocular vision and AprilTag," in *Proceedings of 2018 37th Chinese Control Conference (CCC)*, pp. 4717–4722, Zhuhai, China, 2018.

[11] J. Li, J. A. Besada, A. M. Bernardos, P. Tarrío, and J. R. Casar, "A novel system for object pose estimation using fused vision and inertial data," *Information Fusion*, vol. 33, pp. 15–28, 2017.

[12] E. Olson, "AprilTag: a robust and flexible visual fiducial system," in *Proceedings of 2011 IEEE International Conference on Robotics and Automation*, pp. 3400–3407, Zhuhai, China, 2011.

[13] J. Kalomiros and J. Lygouras, "Robotic mapping and localization with real-time dense stereo on reconfigurable hardware," *International Journal of Reconfigurable Computing*, vol. 2010, Article ID 480208, 17 pages, 2010.

[14] D. Barath and L. Hajder, "A theory of point-wise homography estimation," *Pattern Recognition Letters*, vol. 94, pp. 7–14, 2017.

[15] X. Ren, J. Luo, E. SolowjoW et al., "Domain randomization for active pose estimation," in *Proceedings of 2019 International Conference on Robotics and Automation (ICRA)*, pp. 7228–7234, Zhuhai, China, 2019.

[16] I. K. Park, M. Germann, M. D. Breitenstein, and H. Pfister, "Fast and automatic object pose estimation for range images on the GPU," *Machine Vision and Applications*, vol. 21, no. 5, pp. 749–766, 2010.

[17] S. M. Abbas, S. Aslam, K. Berns, and A. Muhammad, "Analysis and improvements in AprilTag based state estimation," *Sensors*, vol. 19, no. 24, p. 5480, 2019.

[18] C. Song, J. Song, and Q. Huang, "HybridPose: 6D Object Pose Estimation under Hybrid Representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),*, pp. 431–440, Zhuhai, China, 2020.

[19] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proceedings of 2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 292–301, Salt Lake City, Utah, America, 2018.

[20] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.