

Monocular Vision Position Calibration and Manipulator Design

Borong Liang

Beijing University of Civil Engineering and Architecture, Beijing, China

Abstract

A new research application based on the monocular camera estimation technique is presented, which uses the monocular camera OpenMV for object recognition. Mainly, the openMV visual algorithm is used for a positioning, and the number of pixels in the near and far distance is approximately compared, and the distance is accurately measured by some modified algorithms. At the same time, using the deviation of the position in the Angle of view to measure the Angle of the object, it can realize the positioning of the object's spatial coordinates. The coordinate feedback is carried out, and a code is written on the main control component to carry out an inverse kinematics solution, and the Angle of the corresponding manipulator is solved. Then the Angle is fed back to the steering gear, and the mechanical arm can grasp the object accurately. The experimental results show that the motion coordinate estimation result of the system is basically consistent with the expected trajectory result, and it has accurate pose estimation and positioning effect.

Keywords

Monocular Vision; Measuring the Distance; Measuring Angles; Mechanical Arm; Kinematics Forward and Inverse Solutions; Automatic Target Location.

1. Introduction

At present, robots have been widely used in automated production lines, and the demand for visual manipulator is also increasing. Double-sided vision, high price of step harmonic reducer, make in many intelligent production, daily development of students have a certain hindrance. On this basis, a vision manipulator with low price and stable performance is proposed.

2. System Hardware Structure Design

2.1 Monocular Vision OpenMV

Openmv can carry out simple object recognition, color recognition and other features recognition, built-in IC chip, can be directly programmed for machine vision algorithm processing. In the overall implementation process, object identification, ranging and Angle measurement are firstly processed in OpenMV, and then corresponding coordinate values are sent to Arduino Mega 2560 for processing in the form of serial communication. In the process of installation, it is installed in front of the head.

2.2 The Arduino Mega 2560

Arduino Mega 2560 has a variety of serial ports for easy connection of various serial devices. Programming using C language, can have strong computing ability, is conducive to some process algorithm processing calculation.

2.3 The Steering Gear

The freete bus steering gear is adopted, including two kinds of steering gear, sm30 and SM40. The mechanical arm is successively connected by 3D printed materials. In the process of control, the

corresponding Angle rotation can be realized by directly writing Angle, acceleration and velocity. The built-in processing circuit and design, including overcurrent protection, PID algorithm adjustment, further reduce the design difficulty, improve the precision of the manipulator.

3. Design of System Software

3.1 OpenMV Ranging and Implementation

Monocular camera ranging cannot be directly ranging, measuring angles and coordinates. To achieve ranging, it is necessary to select a reference point, which can be determined by itself. Through the selection of reference points, objects can be ranging through mathematical calculation and derivation [1]. Then we'll do a detailed derivation. `blob.w()` returns the width of the color block's outer frame `w(int)`; `blob.h()` returns the height of the color block's outer frame `H(int)`; `blob.Pixels()` returns the pixel number of the color block (`int`). `blob.cx()` returns the center x-coordinate (`int`) of the color block's outer box, and `blob.cy()` returns the center y-coordinate (`int`) of the color block's outer box. We can get some of the values we need by getting the above function. Carefully, it can be found that pixel points can be roughly calibrated by area calculation. In the calculation process, a rectangular frame is used to mark the recognized object, and then using the function `blob.w()` to return the width of the outer frame of the color block `w(int)`, `blob.h()` to return the height of the outer frame of the color block `h(int)`, the area of the object can be calculated, which can be understood as the calculation principle of pixel number. Then, when calculating the relationship between the number of pixels and distance, it is necessary to select a reference distance and then calculate the actual distance by means of proportion [2,3].

Parameter description: reference object pixel A_x , reference point Angle A , focal length L , height, selected reference distance K . The actual object pixel B_x , the actual Angle b , the actual distance L_m , the actual height is H_m , as shown in Figure 1.

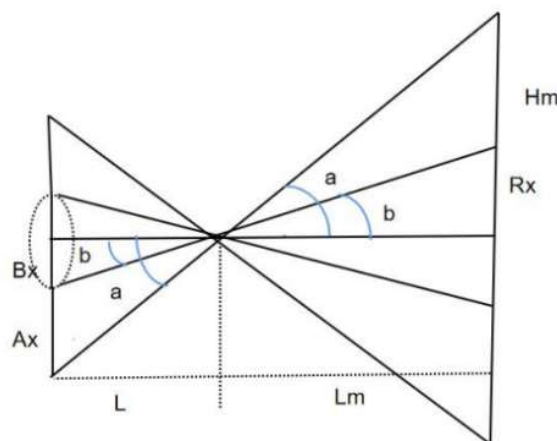


Figure 1. Ranging

Calculation process: $\tan(a) = \frac{A_x}{2L} = \frac{H_m}{L_m}$, $A_x * L_m = 2L * H_m$, $L = \frac{A_x * L_m}{2H_m}$, $\tan(b) = \frac{B_x}{2L} = \frac{R_m}{L_m}$, $B_x * L_m = 2L * H_m$, $\frac{\tan(a)}{\tan(b)} = \frac{A_x}{B_x} = \frac{H_m}{R_m}$, $L_m * B_x = \frac{A_x * H_m}{\tan(a)}$.

L_m times B_x is equal to $L_m = \frac{A_x * H_m}{\tan(a)} \cdot \frac{A_x * H_m}{B_x * \tan(a)}$.

Since A_x , H_m and a are known, they can be determined as constants. Assuming the constant is $K = \frac{A_x * H_m}{\tan(a)}$ can be obtained. The actual distance of $L_m = \frac{K}{B_x}$ can be calculated by calculating the return value of pixels of the actual object.

3.2 OpenMV Angle Measurement and Implementation

In the course of ranging, a calculation has been made for the constant K and the actual distance Lm. Then we'll do a detailed derivation. Using blob.cx() returns the center x-coordinate (int) of the color block's outer box, and blob.cy() returns the center y-coordinate (int) of the color block's outer box. We can get some of the values we need by getting the above function.

Parameter description: Assume that the perspective pixel is 160*120 and the reference object is unbiased at the coordinate of the center of the perspective. The actual object pixel Cx, the actual Angle c, as shown in Figure 2.

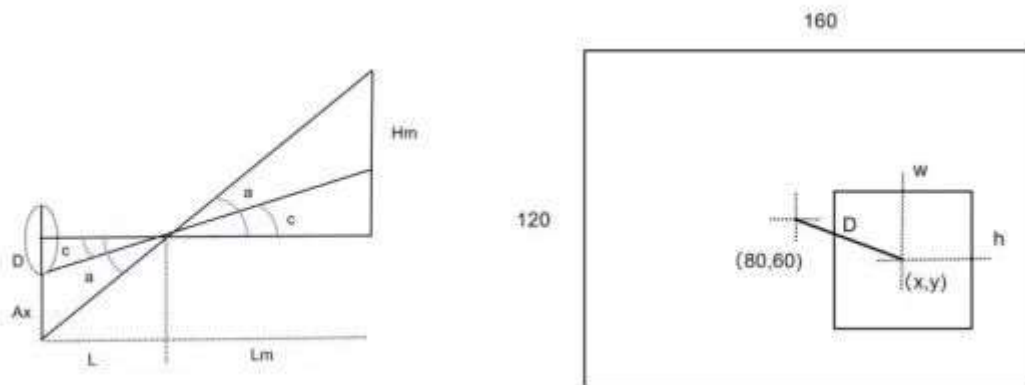


Figure 2. Angle measurement

Through the derivation of distance calculation, we can calculate the actual distance Lm, focal length $L = \frac{Ax * Lm}{2Hm}$.

Calculation process: $D = \sqrt{(x - 80)^2 + (y - 60)^2}$.

$\tan(c) = \frac{D}{L}$, $\tan(a) = \frac{D}{L} \frac{Ax}{L}$, $\frac{\tan(c)}{\tan(a)} = \frac{D}{Ax}$, $\tan(c) = \frac{D}{Ax} \tan(a)$.

$Ax = \frac{Hm}{Lm} * 2L$, $\tan(a) = \frac{Hm}{Lm}$,

$c = \arctan(\frac{D}{Ax} \tan(a))$.

Calculation process: $\tan(a) = \frac{Ax}{2L} = \frac{Hm}{Lm}$, $Ax * Lm = 2L * Hm$, $L = \frac{Ax}{2L} \frac{Hm}{Lm} \frac{Ax * Lm}{2Hm}$.

3.3 OpenMV Coordinate Calculation and Implementation

In the above process, a calculation has been made for constant K, distance Lm, and deviation Angle C. The above calculation results can be used to calibrate the central space coordinates of the object [4]. The detailed derivation process is as follows: The exit deviation Angle C and distance Lm have been calculated. Assuming that the line between the coordinate center of the reference object and the actual object is the Z-axis and the plane of the object is the X-Y plane, the space coordinate calibration is carried out. Where $Z = Lm * \cos(c)$, that's the z-coordinate. Through the above analysis, it will be found that the acquisition of X and Y coordinates will be relatively simple. The coordinates of the object can be easily calibrated using functions blob.cx() and blob.cy(). The center coordinates of the object in the image are (x-60, y-80). The actual coordinate value can be obtained directly through the relationship of proportion. Assuming that the constant of proportion is D, according to the principle of Angle measurement process, the distance amplification constant $d = \frac{Lm}{\cos(c) * D}$, then the corresponding space coordinates will be correspondingly amplified, that is, $X = d * (x - 60)$, $Y = d * (y - 80)$. Through the above analysis and derivation, python is used to write the analysis results into THE

IC of OpenMV, so that the center coordinates of the identified object can be calibrated, and the coordinates can be sent to Arduino for processing through serial communication.

3.4 Forward Kinematics Solution of Five-axis Manipulator

In the process of this design, one axis is to control the opening and closing of the mechanical claw, and the movement is relatively simple, so it is not included in the kinematics equation for analysis. In the process of inverse kinematics solution, the analysis is the four-dOF mechanical arm [5]. The denavit-hartenberg (d-h) model was mainly used for analysis [6,7]. Next, detailed analysis and theoretical calculation are carried out.

Parameter description: install steering gear 0, 1, 2, 3, 4 successively from bottom to top, steering gear 0 as the base of the steering gear, to achieve rotation in XOY plane, the other 4 steering gear in space to achieve movement. Steering gear 1 is directly above steering gear 0 to realize a rotation of space, and the height between is formed on mechanical installation, denoted as d. During production, the lengths of L1, L2 and L3 arms are known as known quantities. Where a0, a1, a2 and a3 are the rotation angles of the four steering gear respectively. The positive solution of motion is the X, Y and Z coordinate parameters obtained from the known Angle of the steering gear, as shown in Figure 3.

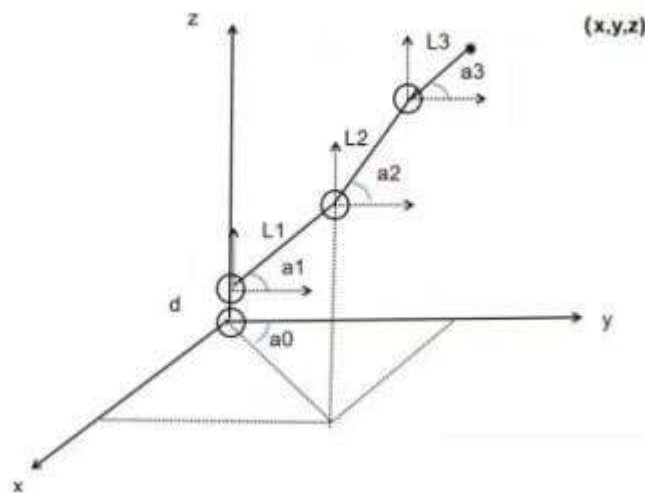


Figure 3. Coordinate parameters

With mathematical knowledge, let's start with a theoretical calculation:

$$\begin{aligned}
 X &= (L1*\cos(a1) + L2*\cos(a2) + L3*\cos(a3))\cos(a0) \\
 Y &= (L1*\cos(a1) + L2*\cos(a2) + L3*\cos(a3))\sin(a0) \\
 Z &= d + L1*\sin(a1) + L2*\sin(a2) + L3*\sin(a3)
 \end{aligned}$$

Through the above formula, in the process of writing the program, only some fixed parameters need to be measured and modified, which can be applied to various mechanical arms.

3.5 Inverse Kinematics Solution of Five-axis Manipulator

In the process of this design, one axis is to control the opening and closing of the mechanical claw, and the movement is relatively simple, so it is not included in the kinematics equation for analysis. In the process of inverse kinematics solution, the analysis is the four-dof mechanical arm [6,7].

Parameter description: The parameters are the same as the positive solution of the motion, please refer to the description above. The inverse solution of kinematics is given X, Y, Z coordinate parameters, to solve the corresponding steering gear Angle. Now let's do a theoretical calculation with some math.

$$x = l_0 \cos \theta_1 + l_1 \cos(\theta_1 + \theta_2) + l_2 \cos(\theta_1 + \theta_2 + \theta_3) \quad (1)$$

$$y = l_0 \sin \theta_1 + l_1 \sin(\theta_1 + \theta_2) + l_2 \sin(\theta_1 + \theta_2 + \theta_3) \quad (2)$$

$$\theta_1 + \theta_2 + \theta_3 \quad (3)$$

The above equations are simplified below. Substitute Equation (3) into Equation (2) and (1), and get:

$$x = l_0 \cos \theta_1 + l_1 \cos(\theta_1 + \theta_2) + l_2 \cos(\theta_1 + \theta_2 + \theta_3) \quad (4)$$

$$y = l_0 \sin \theta_1 + l_1 \sin(\theta_1 + \theta_2) + l_2 \sin(\theta_1 + \theta_2 + \theta_3) \quad (5)$$

For convenience of calculation, let's:

$$d = l_2 \cos \theta_3 - x$$

$$e = l_2 \sin \theta_3 - y$$

Simplify (4) and (5), get:

$$l_1 = (l_0 \cos \theta_1 + d)^2 + (l_0 \sin \theta_1 + e)^2 \quad (6)$$

Through simple operation, it can be solved as follows:

$$\begin{aligned} \sin \theta_1 &= (-b \pm \sqrt{b^2 - 4ac}) / 2a \\ k &= (l_0^2 - l_1^2 - d^2 - e^2) / 2l_1 \\ a &= d^2 + e^2 \\ b &= -2ek \\ c &= k^2 - d^2 \end{aligned}$$

By the same method, θ_2 can be obtained, thus completing the calculation of inverse kinematics. Apparently, eventually there is two correct solutions, as shown by the dotted line in Figure 1. Generally, the dashed line solution is chosen, so that the force on each joint can be smaller. Through the above formula calculation, in the process of writing the program, only some fixed parameters need to be set through measurement modification, can be applied to a variety of mechanical arms.

4. Experimental Device

The experimental device is an experimental platform based on OpenMV and Freeter steering gear.

5. Conclusion

In this paper, the design of the 5-DOF monocular vision manipulator is introduced. Openmv monocular vision is used to guide the manipulator to complete the operation task. The positioning algorithm based on OpenMV writing is used to locate the approximate position of the target in practice and control the manipulator to realize the whole grasping process.

References

- [1] Cao Yang, DING Yimin. Pose Estimation and Target Automatic Positioning System of Manipulator Based on Binocular Vision [J]. Electronic design engineering, 2021, 29 (24) : 107-110 + 115. DOI: 10.14022 / j.i ssn1674-6236.2021.24.022.
- [2] OpenMV embedded image processing. How to use OpenMV range [OL]. (2022,04,16). [not stated date]. <https://book.openmv.cc/image/ranging.html>.
- [3] Ma Qihua, Zhang Chunyan, Ren Hongjuan, Ni Jun, CAI Lingfeng. Monocular OV6620 digital camera in the application of visual range [J]. Journal of mechanical design and manufacturing, 2011 (02) : 71-73. The DOI: 10.19356 / j.carol carroll nki. 1001-3997.2011.02.029.
- [4] Cheng Qian, NIE Zhuo-yun, FANG Hao-cheng, Shao Hui. Experimental system and control method of monocular vision localization for robot tracking [J]. Journal of huaqiao university (natural science edition), 2019,40(04):542-548.
- [5] Lei Penghao. Research on Motion Simulation of 4-DOF Manipulator [J]. Southern Agricultural Machinery, 202,53(01):47-49.
- [6] Hua-wen zhang. 6 dof serial robot kinematics inverse solution algorithm [J]. Journal of foreign electronic measurement technology, 2021, 40 (04) : 53-57, DOI: 10.19652 / j.carol carroll nki femt. 2002493.
- [7] Da Yuesheng, ZHENG Chuyue, Sun Maorong. Mechanical engineering & electronics,2019,37(10):72-75+80.