Research on humanoid robot grasp object

Posraj khadka[*], Yiyang [**],

[*] Control theory and Control Engineering, Nanjing Tech University

DOI: 10.29322/IJSRP.9.12.2019.p9606
http://dx.doi.org/10.29322/IJSRP.9.12.2019.p9606

Abstract- Robot technology has become one of the most core technologies in the competition of overall national strength among countries in the world. Robot Grasping Technology, as one of the most important technologies in the field of Robotics, is the most basic technology. Different from industrial robots, the automatic grasping ability of humanoid robots is mainly affected by the environment and synergistic sensor. What more, the diversity and randomness environment also put forward higher requirements for the sensory system of humanoid robot. In fact, in the real experiments on NAO, the monocular vision is difficult to gather the depth information of the environment; and the overlapping field of view of the binocular vision is small, which limits the grasping scope of NAO. Therefore, the paper presented a solution that robots perceive the outside world through the Kinect senor instead of the camera, and then do a research on the kinematics of the robot, the identification and localization of the target, motion path planning of the manipulator and the real-time object-grasping operation. The main contents of this article are as follows.
(1) According to the robot kinematics, the relevant theoretical knowledge was introduced, and the mathematical model of NAO's arms based on D-H and kinematics forward equations were established, then the inverse kinematics equation was solved by using analytical method, which provides a foundation for the realization of NAO robot grasping target.
(2) For the realization of high-precision target localization of the robot. This paper presented a design scheme of humanoid robot target localization based on Kinect. Firstly, carrying out the image processing of the scene information from Kinect to get the coordinates of the target center point; then the coordinate systems for the Kinect and the robot were established, the Bursa coordinate transformation model between Kinect and the robot was constructed, and the unknown parameters with the model were solved by the linear total least squares algorithm (LTLS) algorithm. After that, to realize the robot target localization, transforming the target center coordinates of Kinect to the robot coordinate system through the model. At last, the experiments were carried out on the humanoid robot NAO, which shown that the proposed target localization is reasonable and feasible; simultaneously, NAO robot can locate the object in real time, reliably with the method, which satisfied the localization accuracy required with the robot to grab the object. It is also more accurate than the monocular vision of NAO robot.
(3) In order to make robots quickly and accurately grasp the object, after analyzing the working space of the robot, the iteration method and the geometric method were used to calculate the motion reachable space of NAO's arms Aiming at the characteristics of multi- joints and high - dimensional space of the NAO robot,
this paper planed the motion path of the robot arm via the improved Rapidly-exploring Random Tree (RRT* algorithm), which introduced the grid search method and the bidirectional expansion strategy into the original RRT* algorithm. The experimental results show that this method not only preserves the characteristics of RRT*, but also improves the efficiency of the path planning.
(4) Under the constraints with obstacles and without obstacle, through the construction of the overall robot self-crawling system, by combining the above (1), (2), and (3) expounded methods, then achieving the goal of the robot autonomy grasps diverse objects in different environments. At the same time, the experiments verify the correctness of the theory used in this paper.
Mention the abstract for the article. An abstract is a brief summary of a research article, thesis, review, conference proceeding or any in-depth analysis of a particular subject or discipline, and is often used to help the reader quickly ascertain the paper's purpose. When used, an abstract always appears at the beginning of a manuscript, acting as the point-of-entry for any given scientific paper or patent application.

Index Terms- Kinect sensor; Target recognition and localization; Motion path planning; the robot NAO; Grasping target

I. Introduction

At present the world is at the confluence of the industrial revolution and the new technological revolution. Continuous innovation has been promoted the vigorous development of robotics technology. Robotics technology is now one of the important technical pillars for development. In order to care for the elder people, for the daily lives of people with disabilities and the liberation for more labor the study of service robots with arms and related technologies is important. Humanoid robot is a kind of service robot that imitates humans in appearance, behavior and language design. The head, arm, torso, and foot are based on machinery and manufacturing technology, and are integrated with communications, control, and computers. With a strong discipline and flexibility, it is suitable for working with humans or replacing human work. Among them, grabbing objects is an indispensable capability for

humanoid robots and is also the most important function of robots. The research of humanoid robots for grasping objects not only includes machine vision technology, but also involves control theory and computers. There are many aspects such as graphics, geometric topology and basic mathematics, so it is challenging to achieve this operational task. further research on humanoid robot technology has extremely important practical significance

II. ROBOT KINEMATICS

A. Computing Forward kinematics

$$ {}^{n}T_{0} = {}^{1}T_{0} \cdot {}^{2}T_{1} \cdot {}^{3}T_{2} \cdots {}^{n}T_{n-1} = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (2.1) $$

Main idea of computing forward kinematics is first use Denavit and Hartenberg D-H method.[1] Specific for each connecting rod to the establishment of coordinate system and then according to the robot each joint variable value of each joint relative to the adjacent joints homogeneous transformation matrix. The final end of the robot kinematics equation is used to derive the actuators relative to the robot base frame ${}^{n}T_{0}$.

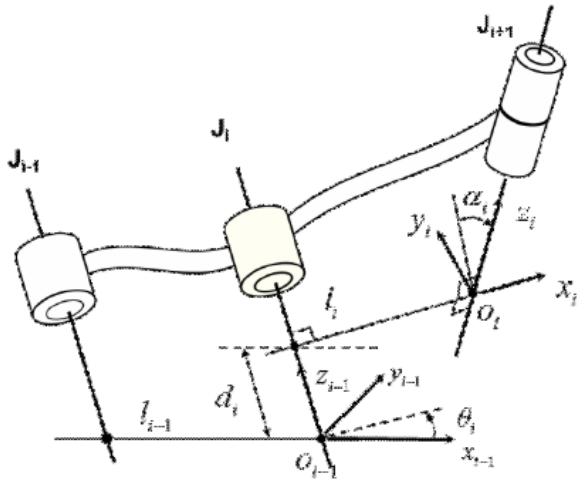


Fig.2.1 Schematic diagram of connecting rod coordinate

For the D-H method is adopted to realize each link coordinate system set up, need to use some of the parameters of each joint and the connecting rod. Which θ_i represents the Angle between the adjacent two connecting rod , d_i for the relative position between adjacent two connecting rods, and l_i normal distance, α_i is perpendicular to the normal plane. The Angle between the two adjacent connecting rod shafts, as shown in figure 2.1 for the connecting rod model. At the same time, this model includes the i - 1, i and i + 1 three joints and connected with the two connecting rods, with the axis of the three joint is normal, each joint with j_{i-1} , j_i and j_{i+1}. Set o_{i-1} For the first i - 1 is connected with the number i-joint connecting rod of the coordinate system origin, x_{i-1} , y_{i-1}, z_{i-1} representatives of link coordinate system of the three-axis direction. o_i number for the i connected to the I + 1 joint of the connecting

rod coordinate system origin, x_i, y_i, z_i for link coordinate system of the three-axis direction. i link for the establishment of coordinate system includes four processes:

1) To determine its position, o_i the origin of the coordinate system. j_i and j_{i+1} has a normal with j_{i+1} of intersection points.

2) x_i shaft , x_i axis and the j_i and j_{i+1} public normal overlap to the direction of j_i to j_{i+1}.

3) Determine z_i axis and j_{i+1} overlap, arbitrary direction

4) Determine y_i axis, after determine the x_i axis and z_i axis direction through the adoption of the right hand rule to get y_i axis direction

$$ {}^{i+l}T_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & l_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & l_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (2.2) $$

If robot joint type belongs to the rotational joint θ_i for the robot movement joint rotation Angle, d_i, l_i , α_i usually have been identified in robot design. So when the robot model with the connecting rod, each joint using the d-h method. Which can obtain the value of the parameter, d_i, l_i, α_i through the type (2.2) can be homogeneous solution of two adjacent joint transform matrix T, (2.1) can be calculated and then reuse the robot actuator at the end of the base coordinates transformation matrix.

B. Computing inverse kinematics

1. Numerical iteration method inverse solution

In solving the robot inverse kinematics solution[2] ,the commonly used numerical iterative methods have Levenberg - Marquardt algorithm (LM) and Newton -- Raphson algorithms.[3] These methods often need to use the Jacobi matrix to solve, because in the process of iterative Jacobi matrix has a clear the direction of search solution and determine the search direction on the function of the distance. Jacobian matrix in the kinematics of the robot is mainly used to describe differential movement and at the end of each joint coordinates.

The equation of motion of robot end actuators are as follows:

$$ x = f(\theta) $$

$$ \frac{dy}{dt} = \frac{\partial f}{\partial \theta} \cdot \frac{\partial \theta}{\partial t} $$

$$ \dot{x} = J(\theta)\dot{\theta} $$

Type, $J(\theta)$ on behalf of the jacobian matrix. When the robot in 3 d space motion, J is 6 x n matrix (n for joint number), if n = 6, J as 6 x 6 phalanx, can be directly on the inverse.

$$ J^{-1}(\theta) = \frac{Adj(J(\theta))}{|J(\theta)|} \qquad (2.1) $$

Adj (J) as the adjoint matrix of $|J(\theta)|$determinant for its value. When J is not square, in the process of the numerical iteration, use the inverse Jacobi matrix should be used when the pseudo-inverse $J^{+}(\theta)$ to carry out operations, and $J^{+}(\theta)$ is expressed as:

$$ J^{+} = J^{T}(JJ^{T})^{-1} $$

J^{T} for transposed matrix. By the $J^{+}(\theta)$ correction for the theoretical basis of the joint variables:

$$d(\theta) = J^{+}(\theta)\,\mathrm{dx}$$

Numerical iterative method is used to analyses the robot inverse kinematics, the need to constantly calculating the jacobian matrix, until achieve a certain set of requirements

2. Analytical method inverse solution

Obviously, by the robot motion equation (2.2) can get n independent equation, and the equation contains all the need to solve the unknown parameters and their solutions. Analytical method was applied to inverse kinematics of the robot, the first direct observation of n independent equation, the equation of only one variable can evaluate first, if the equation containing multiple unknown cannot solve directly, then the robot kinematics equation, using left by a series of transformation matrix inverse (T_i^{-1}) and then analyze each equation to the right elements, with the right end element equation for zero or constant as a breakthrough point, the relative should be on both ends of the equation, then solve the equations. Is widely applied in robot kinematics analytic method for inverse solution of numerical iterative method is mainly for analytic method needs a large number of calculations, requiring higher is not suitable for real time and in the design of the robot structure, usually considering its solvability of the inverse solution. [4]

C. NAO robot model and its inverse solution experiment right arm

Table 2.1 connecting rod parameters of right arm of NAO robot

Frame (joint)	θ_1	d_i / *mm*	l_i / *mm*	α_i^0
Base	A(0)	-ShoulderOffsetY	-ElbowOffset	ShoulderOffsetZ
RShoulderPitch	θ_1	0	0	-90
RShoulderRoll	$\theta_2 + \frac{\pi}{2}$	0	0	90
RElbowYaw	θ_3	0	UpperArmLength	90
RElbowRoll	θ_4	0	0	90
Rotation			$R_z(\frac{\pi}{2})$	
End effector			A(-HandoffsetX-LowerArmLength, 0,0)	
Rotation Fix			$R_z(-\pi)$	

One Shoulder Offset Y = 98 mm, Elbow Offset Y = 15 mm, Shoulder Offset Z = 100 mm, Upper Arm Length = 105 mm, using these parameters, calculate according to the type (2.5) stand out pose transformation matrix ${}^{i+1}T_i$ respectively.

$${}^{1}T_0 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{2}T_1 = \begin{bmatrix} -\sin\theta_2 & 0 & \sin\theta_1 & 0 \\ \cos\theta_2 & 0 & -\cos\theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$ {}^{3}T_{2}=\begin{bmatrix}\cos\theta_3 & 0 & \sin\theta_3 & 0\\ \sin\theta_3 & 0 & -\cos\theta_3 & 0\\ 0 & -1 & 0 & L1\\ 0 & 0 & 0 & 1\end{bmatrix},\quad {}^{4}T_{3}=\begin{bmatrix}\cos\theta_4 & 0 & \sin\theta_4 & 0\\ \sin\theta_4 & 0 & -\cos\theta_4 & 0\\ 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad \text{……. (2.2)} $$

According to the robot kinematics equation, solve the robot actuator at the end of the base coordinate system right arm posture transformation matrix ${}^{0}T_{4}$ as follows.

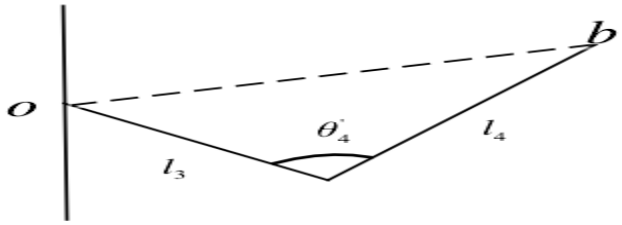
$$ {}^{End}T_{Base}=A_{Base}^{0}{}^{1}T_{0}{}^{2}T_{1}{}^{3}T_{2}{}^{4}T_{3}R_z(\tfrac{\pi}{2})A_4{}^{End}R_z(-\pi)=\begin{pmatrix}n_x & o_x & a_x & p_x\\ n_y & o_y & a_y & p_y\\ n_z & o_z & a_z & p_z\\ 0 & 0 & 0 & 1\end{pmatrix} \text{…… (2.3)} $$

From (2.3) , (2.2) and table (2.1) we will get

$$ o_y=\cos\theta_2\sin\theta_4-\cos\theta_3\cos\theta_4\sin\widehat{\theta}_2 \text{…… (2.4)} $$

D. based on analytic method to solve the NAO right arm kinematics inverse solution

Such type as (2.1), (2.2) and (Shown in 2.3), directly to solve the four joint Angle rotation Angle has certain difficulty, in the process of movement, the robot is composed of upper arm, lower arm and the end executor can form a triangle



(2.2) triangle formed by robot arm

In figure 2.2 shoulder joint o is (s_x, s_y, s_z) and desired location for the end of the actuators, b is (g_x, g_y, g_z). Euclidean distance of two points o,b is

$$ d=\sqrt{(s_x-g_x)^2+(s_y-g_y)^2+(s_z-g_z)^2} \qquad (2.5) $$

According to the law of cosines, find out θ' as follows:

$$ \theta'_4=\arccos\left(\frac{l_3{}^2+l_4^2-d^2}{2l_3l_4}\right) \qquad (2.6) $$

Angle inside the triangle and RELbow Roll joint rotation from zero to positive direction stretch, so RELbow Roll joints of the actual rotational Angle is:

$$ \theta_4=\pi-\theta'_4 \qquad (2.7) $$

Solving the joint Angle of θ_4, then the value in (2.14) in solving other joint Angle, type:

$$ o_y=\cos\theta_2\sin\theta_4-\cos\theta_3\cos\theta_4\sin\theta_2 \qquad (2.8) $$

The range of θ_4 Angle is 0 ~ 88.5 degrees, so:

$$ \cos\theta_3\sin\theta_2=\frac{\cos\theta_2\sin\theta_4-o_y}{\cos\theta_4} \qquad (2.9) $$

Another：

$$p_y = g_y = -l_1 - l_3\cos\theta_2 - l_4(\cos\theta_2\cos\theta_4 - \cos\theta_3\sin\theta_2\sin\theta_4) \quad (2.10)$$

By type (2.9) type (2.10) can be calculated:

$$\theta_2 = \arccos\left(\frac{-g_y - l_1 - (\frac{l_4\sin\theta_4 o_y}{\cos\theta_4}}{l_3 + l_4\cos\theta_4 + l_4\frac{\sin^2\theta_4}{\cos\theta_4}}\right) - \frac{\pi}{2} \quad (2.11)$$

According to the type (2.11):

$$a_y = \sin\theta_2\sin\theta_3 \quad (2.12)$$

The value of θ_3 as follows:

$$\theta_3 = \arcsin\left(\frac{a_y}{\sin(\theta_2 + \frac{\pi}{2})}\right) \quad (2.13)$$

When $\theta_3 \neq \frac{\pi}{2}$, According to (2.13) the value of θ_1 is:

$$\theta_1 = \arccos\left(\frac{a_y + \frac{n_x\sin\theta_3\cos\left(\theta_2 + \frac{\pi}{2}\right)}{\cos\theta_3}}{\cos\theta_3 + \frac{\cos^2\left(\theta_2 + \frac{\pi}{2}\right)\sin^2\theta_3}{\cos\theta_3}}\right) \quad (2.14)$$

When $\theta_3 = \left|\frac{\pi}{2}\right|$,According to (2.14) the value of θ_1 is:

$$\theta_1 = \arcsin\left[\frac{a_x}{\cos(\theta_2 + \frac{\pi}{2})\sin\theta_3}\right] \quad (2.15)$$

This paper uses the reverse thinking, using the method of robotic forward kinematics for testing. First of all, within the scope of the NAO right arm rotation of each joint, freely choose 8 groups of joint variables as the test sample, as shown in table 2.16. Then according to the kinematics of a robot is corresponding to the transformation matrix of all sample. To verify the reliability and accuracy of this method.

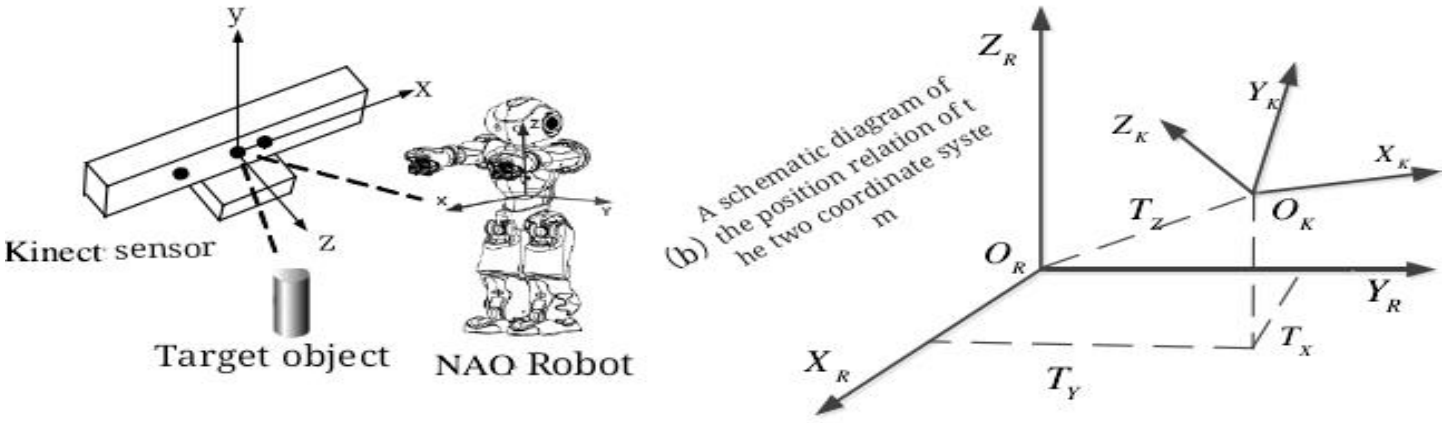
Table 2.2 sample data of each joint angle of right arm

sample	RShoulderPitch(^{0})	RShoulderRoll(^{0})	RElbowYaw(^{0})	RElbow(^{0})
1	20.34	-60	-64.25	0.00
2	45.89	-18.43	-0.61	13.24
3	-2.39	-1.85	-100.52	2.09
4	-56.71	10.00	102.00	41.25
5	3.48	3.94	9.34	9.24
6	100.52	-15.74	-45.78	17.23
7	-95.27	-20.99	33.26	40.54
8	3.06	5.46	-59.04	57.87

E. Kinect based robot target positioning.

Kinect get the scene RGB-D image. Converting RGB space in HSV space [5-6]. Drawing H-S color histogram and at last determining threshold segmentation range based on the histogram.

The establishment of the device coordinate system, as shown in figure below, the origin is located in the RGB camera, the x axis by origin from left to right, y-axis vertical main axis up, according to the right department law determine the z-axis. NAO robot was set up with 3 sets of the base coordinate system[7]: the world coordinate system (FRAME_WORLD), the robot coordinate system (FRAME_ROBOT) as well as the body coordinate system (FRAME_TORSO).



(a) Kinect and NAO three-dimensional coordinate system

(b) Schematic diagram of the positional relationship between Two coordinate systems

Fig 2.3 Coordinate system establishment for Kinect and NAO

The 3-d coordinate transformation model and solve the parameters. From the above fig. there are three translation quantity. There is also a large Euler angle between the axis. As long as rotational translation can make two coordinate systems. we used one of the most widely used coordinate transformation model for seven parameters of the model. [8-9]

$$\begin{bmatrix} X_R \\ Y_R \\ Z_R \end{bmatrix} = (1+\lambda)R(\in)\begin{bmatrix} X_K \\ Y_K \\ Z_K \end{bmatrix} + \begin{bmatrix} \Delta T_x \\ \Delta T_y \\ \Delta T_z \end{bmatrix}$$

Type, $\left[X_K Y_K Z_K\right]^T$ point coordinates in the device coordinate system $\left[X_R Y_R Z_R\right]^T$ for the coordinates of points in the robot coordinate system; The scaling factor lambda for dimension; As [Δ Tx Δ Ty Δ Tz] ^{T} said coordinate translation; R (∈) as the rotation matrix and R (∈) = [R (∈ x) R (∈ y) R (∈ z)]. Two corresponding Euler Angle coordinate system; As definition dx = (Δ Tx Δ Ty Δ Tz $\in_Z \in_Y \in_X$ (1 + Δ))

For vector coordinate transformation parameters. In solving coordinate conversion model to solve the dx seven unknown parameters, and it usually takes three or more than three common points with two sets of coordinates. As a result of the existence of common point inevitable error, the introduction of coordinate correction, will get type type (3.8) (3.9) as shown in the error equation

$$e_{3n\times1} = A_{3n\times7} \cdot dx_{7\times1} + l_{3n\times1}$$

Type: n common point number; A for the coordinate transformation matrix. ℓ for observation vectors for the two-coordinate system.

$$l_{3n\times1} = \left[x_K - x_R \quad y_k - y_R \quad z_k - z_R\right]^T$$

When n > 3, the coefficient matrix A often exist error, will introduce the total least squares algorithm model, as shown below:

$$e - l = (A - E_A) \times dx \qquad (2.16)$$
$$(rank(A) = m \square \ n)$$

The error vector e, E_A for error matrix of coefficient matrix A, for solving the parameter vector dx, the key is the E_A reduction effect matrix A, but the direct solution has the certain difficulty. Therefore, this article adopts linear total least squares algorithm (LTLS) type for linear processing, can simplify the calculation process and to ensure conversion precision[10] . At this point, provide the n observer $L_1, L_2, L_3 .... L_n$ in order to estimate t an independent variable.

$$\begin{aligned} L_1 &= L_1 + V_1 \\ L_2 &= L_2 + V_2 \\ &\cdots\cdots\cdots\cdots.. \\ L_n &= L_n + V_n \end{aligned} \qquad (2.17)$$

Type of V_i for the remaining vector observer, if n observers are linear model:

$$\begin{aligned} &F_1(x_0 + \zeta)L_1 + F_1(x_0 + \zeta)V_1 + F_2(\mathrm{x}_0 + \zeta)L_2 + F_2(x_0 + \zeta)V_2 \\ &+ ..... + F_n(x_0 + \zeta)\mathrm{L}_n + F_n(x_0 + \zeta)V_n + F_0(x_0 + \zeta) = 0 \end{aligned} \qquad (2.18)$$

X_0 for variable is obtained by nonlinear total least square algorithm approximation, the approximate correction factor said had estimated when $F_i(x_0 + \zeta)$ is a nonlinear equation, using Taylor's series of X_0 first derivative:

$$\begin{aligned} &F_1(x_0)L_1 + F_1^{'}(x_0)\mathrm{L}_1\zeta + F_1(\mathrm{x}_0 + \zeta)\mathrm{V}_1 + \\ &F_2(x_0)\mathrm{L}_1 + F_2^{'}(x_0)\mathrm{L}_2\zeta + F_2(\mathrm{x}_0 + \zeta)\mathrm{V}_2 \\ &\qquad\vdots \\ &\qquad\vdots \\ &F_1(x_0)L_1 + F_n^{'}(x_0)\mathrm{L}_n\zeta + F_n(\mathrm{x}_0 + \zeta)\mathrm{V}_n + \\ &F_0(x_0) + F_2^{'}(x_0)\zeta \end{aligned} \quad =0 \qquad (2.19)$$

$$\begin{array}{ccc} \downarrow & \downarrow & \downarrow \\ L & A\zeta & BV \end{array}$$

Type (3.14) is another form of equation (3.8), and contains all observer residuals. When all the observer is independent, objective function can be written as:

$$\phi = V_1^T P_1 V_1 + V_2^T P_2 V_2 + \cdots + V_n^T P_n V_n - 2K^T(A\zeta + BV + L) = \min \qquad (2.20)$$

P for the observed value of weight matrix, the Lagrangian function of m x 1 K said vector, finally through the euler - Lagrange necessary condition to get the necessary equations as follows:

$$\frac{\partial \phi}{\partial V_1} = 0, \frac{\partial \phi}{\partial V_2} = 0, ... \frac{\partial \phi}{\partial V_n} = 0; \frac{\partial \phi}{\partial \phi} = 0 \qquad (2.21)$$

The equation (2.19), (2.20) and equation (2.21) integration:

$$dx = \left[A^T P_i (\sum_{i=1}^{n} F_i^2 (x_0 + \zeta))^{-1} A)^{-1} \right] \bullet \left[A^T P_i (\sum_{i=1}^{n} F_i^2 (x_0 + \zeta))^{-1} \mathrm{L})^{-1} \right] \qquad (2.22)$$

According to the type (2.21) can solve the dx parameter vector can be obtained according to the type (2.22) out of all the value of the parameter vector dx, so the coordinate transformation model is able to solve. All the values, so the coordinate transformation model is able to solve.

(3) The robot localization grab points

By LTLS algorithm to solve the coordinate transformation model to establish a linear relationship between two sets of coordinate system, the device coordinate system according to the right-hand rule x axis rotate - PI / 2, and then around the z axis rotation Angle of PI / 2. The rotation after two, two corresponding coordinate axis Angle with small Angle linear relations, transition matrix T as follows:

$$T = R_x(-\frac{\pi}{2)} R_z(\frac{\pi}{2}) \qquad (2.23)$$

$$\begin{bmatrix} X_K^{'} & Y_K^{'} & Z_K^{'} \end{bmatrix} = T \begin{bmatrix} X_K & Y_K & Z_K \end{bmatrix}$$

$$= \begin{bmatrix} -Z_K & -X_K & Y_K \end{bmatrix} \qquad (2.24)$$

Image processing through access to information and access to common point in device coordinates of the initial coordinate values for $\begin{bmatrix} X_K Y_K Z_K \end{bmatrix}^T$ again by type (2.22) type (2.23) for coordinate transformation to calculate the actual public point coordinates $\begin{bmatrix} X_K^{'} Y_K^{'} Z_K^{'} \end{bmatrix}^T$ .The coordinates of the common point in the robot coordinate system $\begin{bmatrix} X_R Y_R Z_R \end{bmatrix}^T$ by robot position sensor and precise measurement. When the device and robot coordinate corresponding direction axis Euler Angle for small Angle:

$$\sin \in \approx \in, \cos \in \approx 1$$

$$R(\in) = \begin{bmatrix} 1 & -\in_z & \in_y \\ \in_z & 1 & -\in_x \\ -\in_z & \in_x & 1 \end{bmatrix} \qquad (2.25)$$

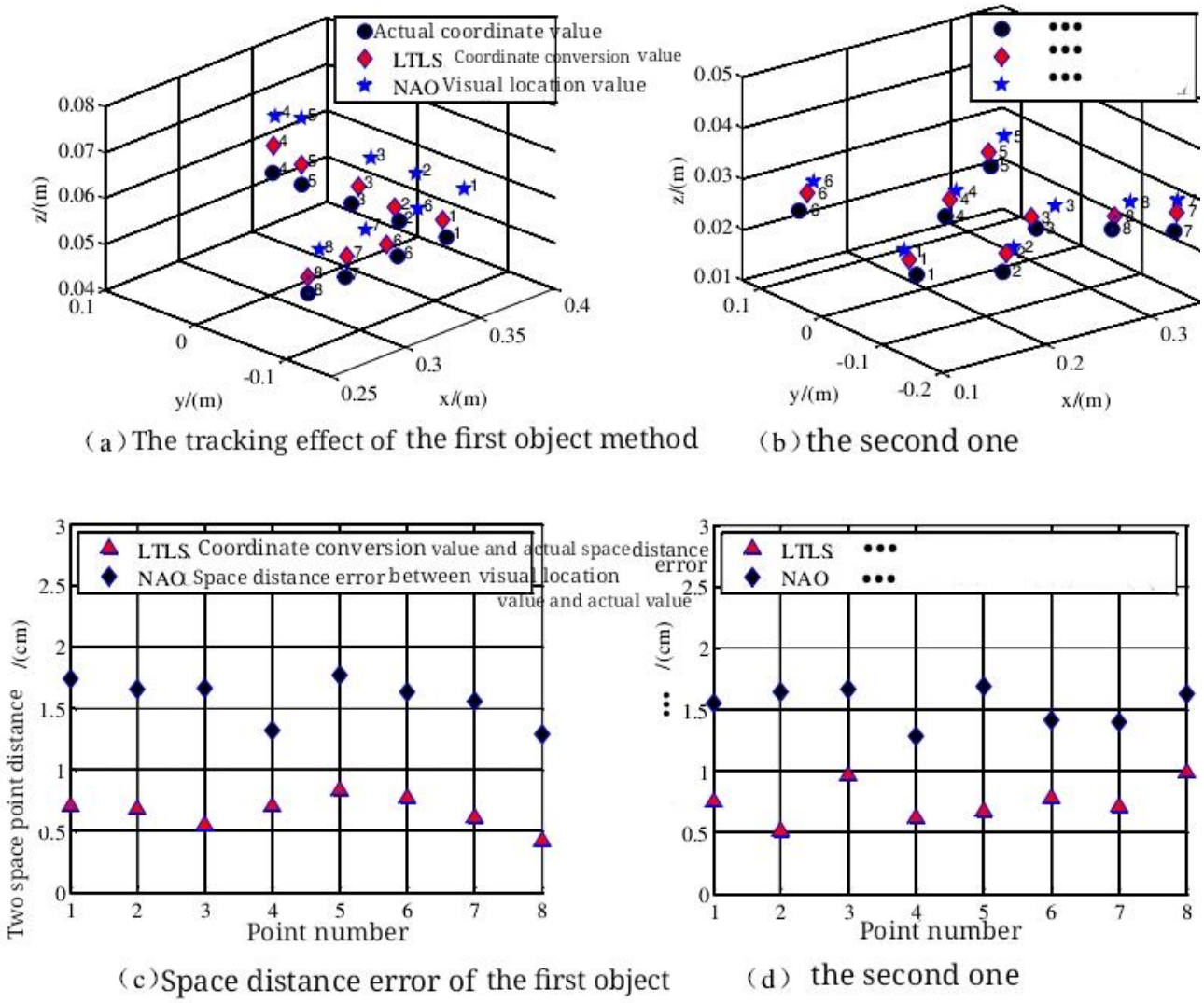
$$P_i (\sum_{i=1}^{n} F_i^2 (x_0 + \zeta))^{-1} = E$$

By type (3.8) type (3.9) can be coefficient matrix A is:

$$A_{3n \times 7} = \begin{bmatrix} 1 & 0 & 0 & 0 & -z_{k1} & y_{k1} & x_{k1} \\ 0 & 1 & 0 & z_{k1} & 0 & -x_{k1} & y_{k1} \\ 0 & 0 & 1 & -y_{k1} & x_{k1} & 0 & z_{k1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & -z_{kn} & y_{kn} & x_{kn} \\ 0 & 1 & 0 & z_{kn} & 0 & -x_{kn} & y_{kn} \\ 0 & 0 & 1 & -y_{kn} & x_{kn} & 0 & z_{kn} \end{bmatrix} \qquad (2.26)$$

The first scene of the sensor device to collect information for a series of image processing and recognition for coordinate transformation common point coordinate positioning and target grasping point coordinates; Then will be two rotating device

coordinate system transformation to get the new device Coordinate system, and then obtain common point and target fetching in the new system of coordinates and common point coordinates in the robot system; Finally, using the LTLS algorithm of coordinate conversion parameters and according to the coordinate transformation model will be called target grasping point coordinate transformation under the new coordinate system to the machine Coordinate system, when the coordinate transformation value and the error between the actual value is less than 1 cm then completed the robot grasping point positioning otherwise repeat the above process.



(a) The tracking effect of the first object method (b) the second one

(c) Space distance error of the first object (d) the second one

Fig.2.4 experiment result

Aiming at the shortcomings of the traditional visual sensor exists, use it depth sensor camera instead of a humanoid robot ontology, this paper proposes a humanoid robot target positioning system based on access, information provided by the device scene make a series of image processing, segment the target object, and realized to the center of the target object positioning. By building access and robot coordinate system and the transformation model using linear total least squares (LTLS) algorithm to solve the model then Device coordinates of target capture under accurate point coordinate to robot coordinate, so as to realize the robot positioning objects. In humanoid robot NAO platforms, experiments have been carried out to verify this method and analyzed the deficiency of this method. The results show that within a certain space, when using this method for target localization, can meet the required the

positioning accuracy of robot grasping objects, positioning accuracy is higher. The NAO monocular vision again to grab objects for subsequent robot operation.

III. Motion path planning algorithm based on improved RRT*

Motion path planning refers to planning a plan according to a specified standard under certain constraints. The path from the initial state to the target state collision avoidance. According to which the robot can avoid obstacles and perform task operations. According to the planning space can be divided into motion path planning under space and Cartesian space. Among them, although the motion planning in the joint space has better real-time performance, it lacks an intuitive path planning process; In Cartesian space, it is necessary to convert the pose information into a joint angle which is computationally intensive but in an obstacle environment. This space is easier to plan for joint space when panning a path. The robot may be subject to physical constraints and obstacles when performing the grab operation etc. Where the physical constraints are mainly refers to the limitations of the model of the robot itself such as, the length of the link and the range of joint motion. Obstacle constraint refers to the operation. There are objects in the workspace that prevent the robot from performing the grab. Therefore, when studying the target of humanoid robots. By defining the workspace and storing the reachable area information of the robot can avoid doing nothing outside the constraints. Traditional motion path panning algorithms include artificial potential field method, grid method, viewable method and topological method. As the robot joints continue to increase the amount of calculations is also the exponential relationship. Therefore, the traditional method is difficult to meet the multi-joint robot for collision avoidance path planning in complex environments. However, the sampling-based motion path planning algorithm refers to the collision avoidance of sampling points in the workspace. Crashing to find a feasible path, avoiding the specific construction of the task space, reducing the amount of calculation and path planning can also be effectively performed in high-dimensional apace and complex environments. Therefore, this paper selects the sampling-based planning algorithm for the machine. The scene capture process performs motion path planning. In the cartesian space, the NAO robot is sitting on the torso. Marked as the base coordinate system, the iterative method and geometric method are used to solve the reachable space range of the right arm of the NAO robot. And through MATLAB for simulation experiments; then summarize several sampling planning methods. The deficiencies of the algorithm are proposed and the improved RRT* algorithm is used to plan the motion path of the humanoid robot arm.

The Rapidly Random Tree method (RRT) [11-12-13] means that in the workspace, the initial nodes are continuously connected by random sampling. The sampling point in free space until the position near the target point or the target point is queried, and the free sampling is performed by connecting the initial point. The algorithm can not only solve the problem of multi-degree-of-freedom robot path planning problem and also guarantee the speed of planning and the quality of the path, the computational complexity will not increase with the complexity of the environment nor use a road map in advance. Therefore, this paper selects this algorithm for robot grasping motion path planning.

Lozano Perez and Wesley state Space was put forward for the first time in 1978 (the Configuration Space), referred to as C_{space} [14] Convenient for the structure of the planning implement and then gradually put forward the C_{free} free Space and $C_{obstacle}$ Obstacle Space concept. In the space the robot is represented as a point. path planning for robot arm movement then converted into a point for the path planning. In the plane C_{space} , the basic RRT algorithm extended process as shown in the fig below.

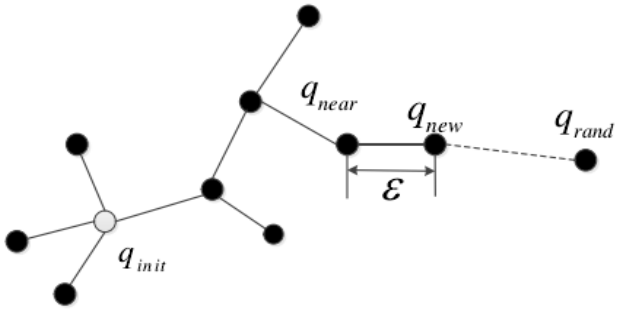


Fig. 3.1 The RRT algorithm search processes

Defines C_{free} without collision free space, C_{obs} to contain obstacle space, q_{init} as the starting point of the initial state, q_{goal} to end point target state, namely q_{init} and q_{goal} are C_{free} .In a bit. T_n for RRT containing n nodes, and $T_n \in C_{free}$. Random sampling point

in C_{space} for q_{rand}, is only effective in C_{free} q_{rand}, after by setting RRT search step length, find the distance q_{rand} recent q_{near} leaf node, and $q_{near} \in T_n$. According to the Euclidean distance, define D (x1, x2) said the x1 state to x2 distance, one of them X1, x2 $\in C_{free}$, usually D (x1, x2) less than or equal to the RRT step length. From q_{near} on q_{near} and q_{rand} direction vector to D (x1, x2) = epsilon for conditions. To explore a new node q_{new} if q_{new} exists and $q_{new} \in C_{free}$ is to expand the node is added to the tree of Tn and Tn to Tn + 1. If $q_{new} \in$ Cobs, select q_{rand} again, repeat the extension steps, Until the search out the finish item standard or radius is P, target for the circle in the center of the node which is suitable for a small fixed P value. q_{new} calculation formula is:

$$q_{new} = q_{near} + \rho * (q_{rand} - q_{near}) / \left\| q_{rand} - q_{near} \right\|$$

Use of RRT algorithm in [0, 500] x [0, 500] barrier-free plane space, iteration 50,200,500,1000,1500,2000 from step as shown in figure 4.2.

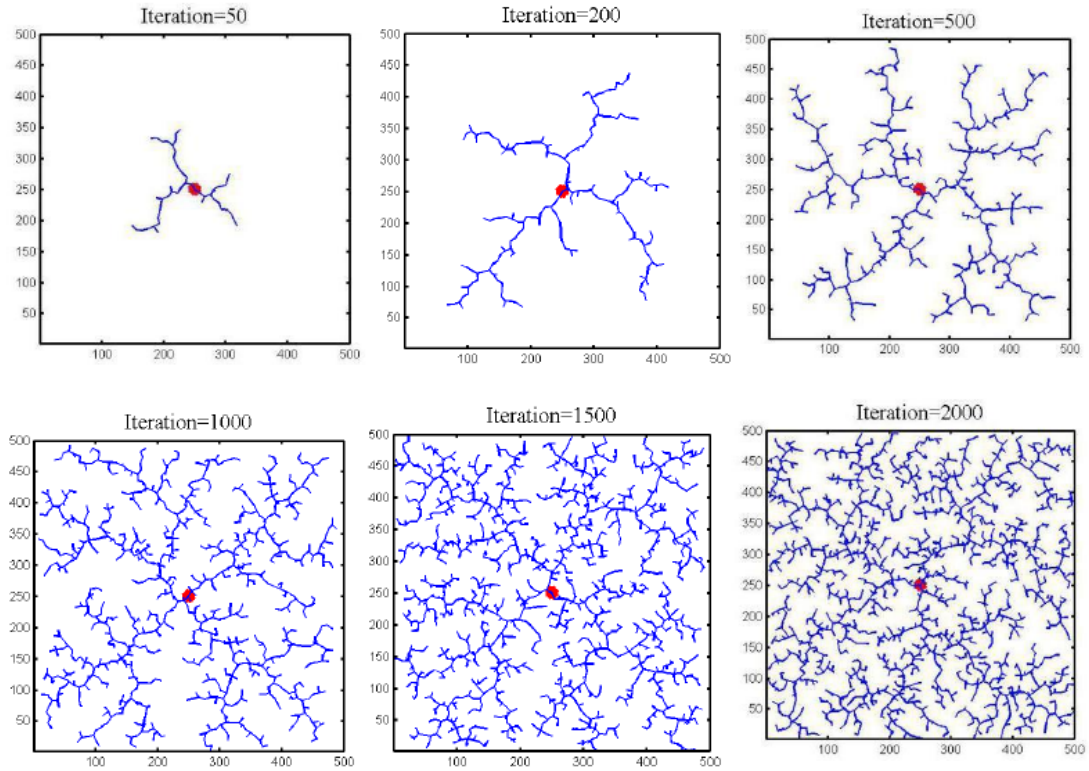


Fig. 3.2 Experiment of the RRT algorithm

Can be seen from the figure 3.2, with the increase of the number of iterations, constantly extend in the state space not search tree node and the subject branches extend four vertices. When after reaching a certain number of random sampling point, expand the tree will be throughout the whole state space, which reflects the RRT algorithm has the characteristics of probability and integrity and show the extension node consistency with the distribution of the random sampling points. At the same time, due to the expansion of RRT algorithm by detecting movement process node whether collision, without the need for obstacles to specific modeling in state space, improve the efficiency of the motion planning. RRT algorithm also has to solve high-dimensional space characteristics of the motion planning problem. Although the algorithm has many advantages and researchers widely used, but it also has some disadvantages. Due to the extended tree is based on random sampling points generated in state space So easy to cause when performing the same task every path of planning are not identical, to reduce the repeatability of operation; Obtained by random sampling points at the same time the randomness of extension path, and sometimes get path is close to the optimal path planning, sometimes deviate from the optimal path, and the convergence speed is slow.

At present, aimed at the shortcoming of RRT algorithm mainly from two aspects of sampling strategy and expanded form is improved[15-16]. An improved algorithm based on sampling strategies mainly include: set the target for the sample point planning Goal - Bias algorithm, this algorithm also can design the number of occurrences of target in the search process. Goal - through around the target increase sampling frequency Zoom algorithm to improve the speed of convergence to the target point etc. Extension form

improvements include improving the convergence rate of bias RRT_Connect extended tree, two-way Random extension algorithm (or called Bidirectional Rapidly - Exploring the Random Trees, Bi - RRT) and optimize the path of the quality of RRT * algorithm.

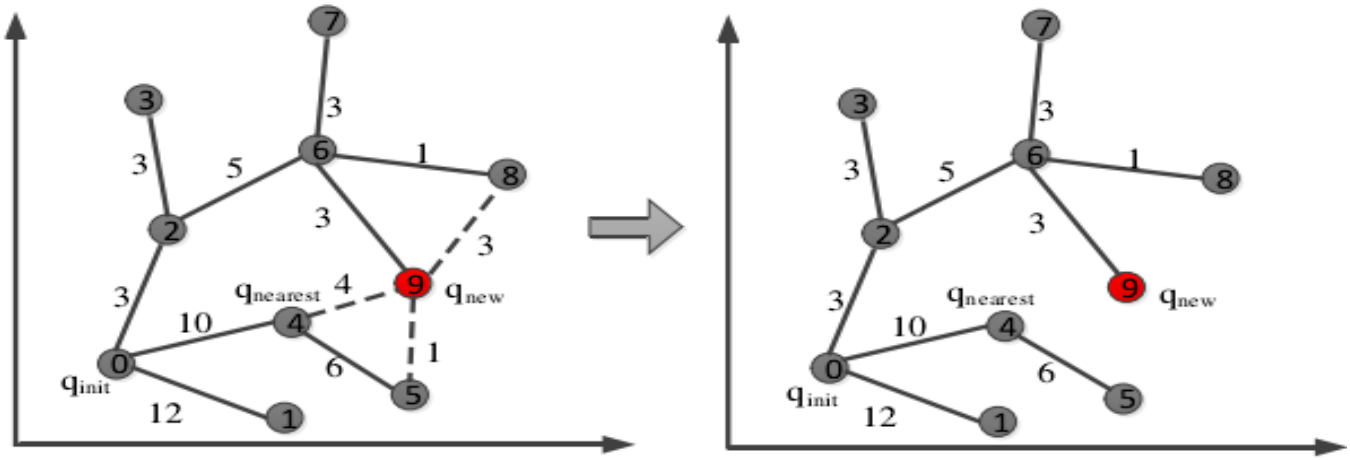
(1) Two-way random extended tree algorithm (Bi-RRT or RRT_Connect)

Due to the many obstacles or high degrees of freedom robot cases, the basic RRT algorithm will spend a lot of time for sampling and detection of collision avoidance, reduce the planning efficiency. Therefore, Kuffner presents a two-way random extension RRT algorithm, namely RRT_Connect algorithm. This algorithm and the basic RRT algorithm mainly has two difference: 1) the initial state points q_{init} and target terminate q_{goal} two trees extension structure at the same time, in the process of iteration, the application of Swap (T1, T2) functions, the two trees extend towards each other, until the two meet, so the algorithm is a heuristic search and not as random as basic RRT algorithm search. 2) The Connect () function (T, q) the introduction of each loop in q_{near} exploring and q_{rand} direction vector multistep to detect obstacles is different from the basic RRT algorithm. only explore step improve the speed of tree extension.

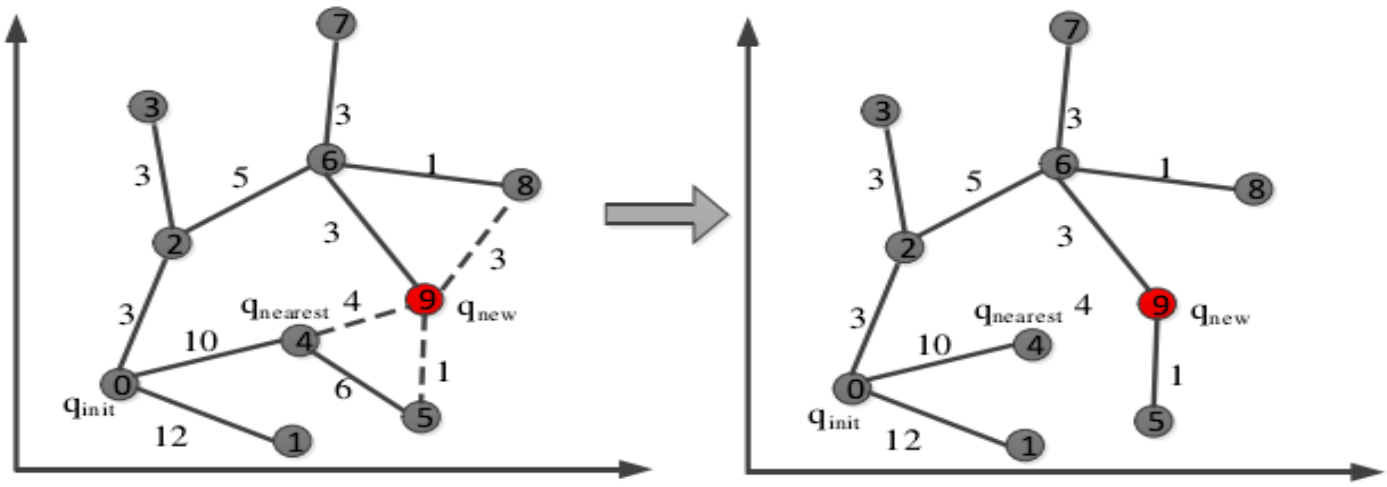
*(2) The RRT * algorithm*

The RRT path planning algorithm randomness makes planning often deviate from the optimal path as a result in improving the quality of path planning. which made certain contribution to many researchers. Sertac and Emilio RRT * algorithm [17] (Rapidly - exploring the Random Tree Star) to ensure that the planning path gradually tend to be optimal and analyzed from the Angle of the theory of how to be optimal and verified RRT.

RRT * algorithm in pseudo code, Cost (q) has been search tree calculated from the initial point to the extension point q path cost. Each extension is a new node, during this period need to figure out what is cost price. RRT * algorithm and obvious difference is the basic RRT algorithm. There is no direct q_{new} with the newly generated node q_{near} connected but do the processing of two aspects: 1) the corresponding algorithm 3, ChooseParent () function modules: the new node near q_{new} Qnear, explore to Cost (q_{new}) price Cost minimum and use it as a q_{new} Parent node, to ensure that the new extension node q_{new} with local costs at a minimum Cost figure 4.6 (a) shows the improvement of schematic diagram; 2) in the corresponding Algorithm3 ReWire () function modules: when the q_{near} nodes near after a new node q_{new} the path cost cost less than the price of before q_{near} . Will replace for q_{near} to q_{new} node, as shown in figure 4.3 (b) schematic diagram for it.



(a) RRT * algorithm first optimization



(b) RRT*algorithm second optimization

Fig. 3.3 The two optimization steps of the RRT* algorithm

*(3) Improved RRT * algorithm*

RRT*algorithm is presented the basic RRT algorithm to improve the quality of the path planning but still a planning efficiency is low. The defects of slow convergence to the optimal path. Two-way random extended tree algorithm is the basic RRT algorithm improves the efficiency of path planning. This paper introduces the search strategy to RRT *, the grid application is fast search algorithm at the same time improved further make the planning efficiency.

In this article the main ideas of grid fast search algorithm include two points:

(a) the division of grid, due to the multiple grid division will bring the double increase of time and space complexity, so only here on a grid. In plane state space, first of all, according to the scope of the work area to determine partition set MinX, MaxX ,MinY, MaxY to x and y direction on the range of minimum and maximum range (in the three dimensional state space in the same way to increase the z axis), set up two axes divided step for k, k > r, r q_{new} find nearby node set distance, Among them:

$$m = \frac{MaxX - MinX}{K}$$

$$n = \frac{MaxY - MinY}{K}$$

(b) To calculate the position of all sampling points and extension points in the grid and recorded. In the new node q_{new} and node q_{near} nearby, just under the grid search rather than traverse the r Within the scope of all the nodes. The grid search algorithm diagram as shown in 4.4.

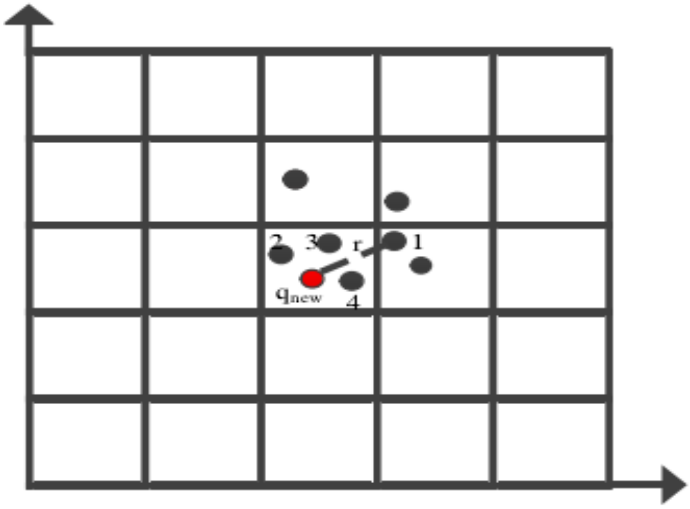


Fig. 3.4 The schematic diagram of grid search method

Extension of the new node q_{new} directly to the vertex 2, 3 and 4 as node Q_{near} recently, and no longer will the vertices 1 as the nodes near the next price cost calculation comparison, thus reducing the cost in the process of trajectory planning time, improve the search efficiency.

IV. Experimental comparison and analysis

This section mainly under the plane space and three-dimensional space for robot path planning algorithm for validation because only care about the positions of the robot in the plane, so the robot as a point. An obstruction in the simulation environment, by comparing the basic RRT algorithm, the original RRT * algorithm and improved RRT * algorithm and the programming efficiency so as to verify the improved RRT * reliability and validity of the algorithm.

Plane state space selection range of 40 x 40 inside the space Settings as shown in figure 4.9Shown in the two irregular quadrilateral as obstacles, on the right side of the barrier of vertex set to {(5.27, 5.27), (0.18, 0.18), (2.98, 2.98), (7.96, 7.96)}, the left obstacles vertex set to {(4.25, 0.025), (12.89, 5.18), (4.01, 8.48), (0.58,1.44)}. q_{init} initial node and destination node q_{goal} (15, 5.5) and (8, 3.65) respectively. Max-step = 0.5 , find nearby nodes radius r = 1.5, $delta\_p_{goal}$ =1 . Figure 5.1 shows the basic RRT algorithm, the original RRT * algorithm and improved RRT*. Without collision algorithm under different number of iterations planning path graph. Among them, the black irregular are obstacles, green for extended tree leaf nodes on behalf of the search process and Blue is from the initial point to the target point path planning.

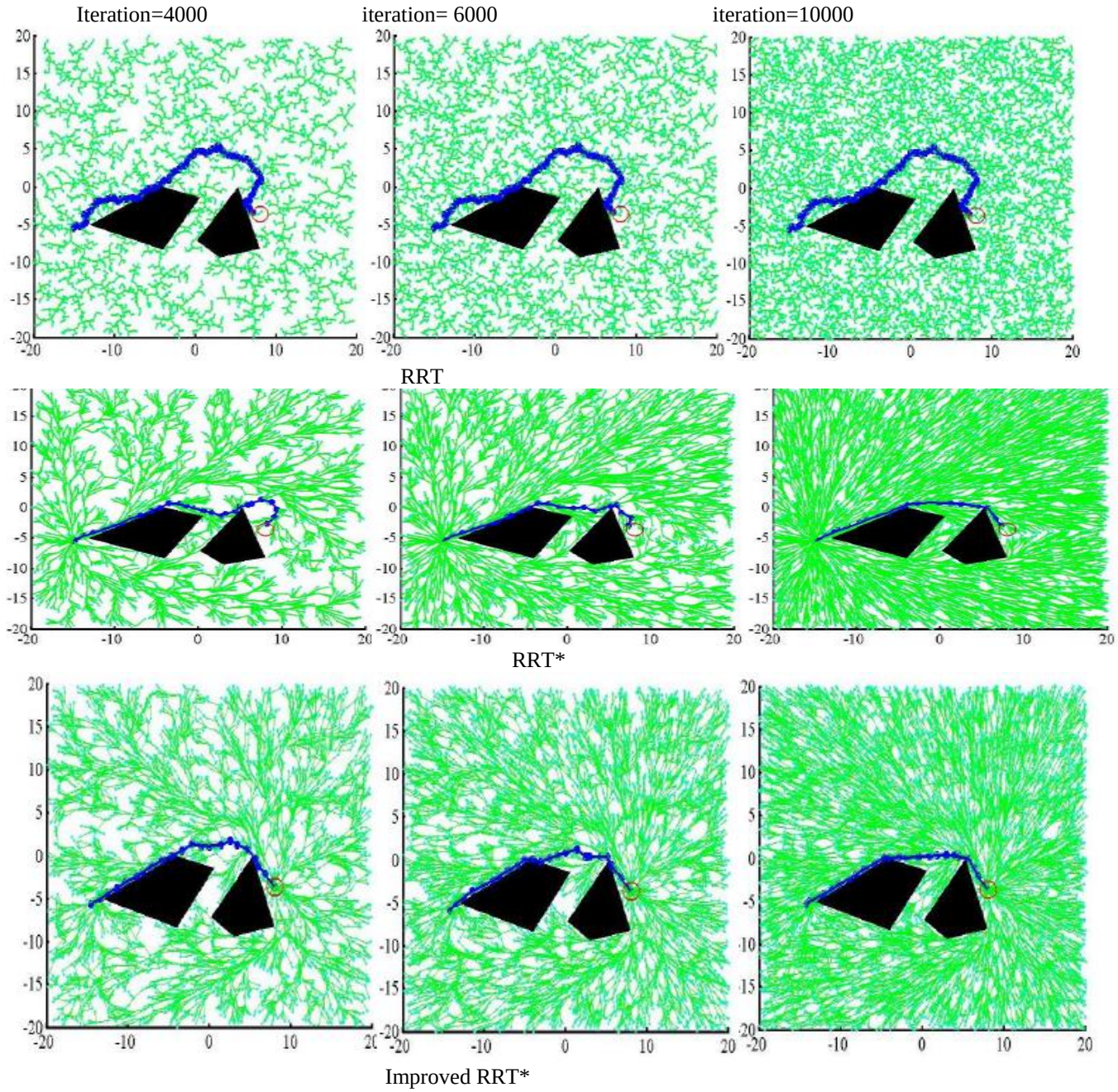


Fig.4.1 Path planning of different algorithms in 2D plane

As you can see from figure 5.1 winding basic RRT algorithm, deviating from the shortest path, and with the increase of the number of iterations path has not been optimized. With the increase of iteration times in RRT * algorithm the basic RRT path planning gradually tend to be optimal, and the improved RRT *. Planning algorithm is more primitive RRT * algorithm path a little advantage. In figure 4.1 environment, the number of iterations is seven thousand times (stable state the number of iterations), 50 on three different algorithms respectively. Time simulation, calculate the iterative search path in the process of planning, cost, and tend to be optimal. Path used by the average of the time, as shown in table 4.1. Along with the increase of the number of iterations, the three different algorithms converge to the optimal path of trend as shown in figure 4.2.

Tab 4.1 The average sampling points and planning points of the three algorithms in plane space

Algorithm	The average path planning point (a)	The average price cost	The average time (s)
The basic RRT	79	42	38.768
RRT * algorithm	26	26.953	32.343
The improved RRT * algorithm	22	24.475	11.446

From table 4.1, the original RRT * algorithm than the average number of path planning point of basic RRT algorithm significantly reduced, path quality has improved, but the path to the optimal value of the time difference; And the improved RRT * algorithm on path planning efficiency obtained significantly higher, on average spending more primitive path planning point and average price RRT * algorithm is also reduced slightly.

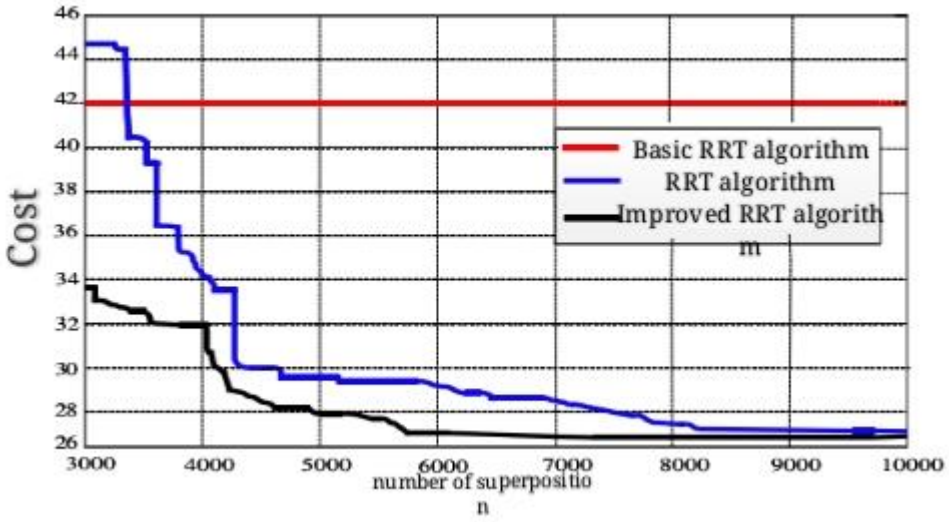


Fig.4.2 Three different algorithms converge to the optimal path in Plane space

As can be seen from the figure 4.2, with the increase of the number of iterations, basic RRT algorithm program did not get any improvement in the quality of the path and use RRT * * planning algorithm and the improved RRT path are gradually tends to the optimal path; At the same time, due to the improved RRT * algorithm introduces the bidirectional search strategy and the grid search algorithm quickly, make it has dramatically improved the efficiency of path planning.
In [0100] * [0100] [0100] of the simulated in three dimensions as shown in figure 4.11 the four obstacles, obstacles radius to 10. Selection of initial point q_{goal} ,10,10 (10), target q_{goal} ,90,90 (90), from the initial point to the target point, set extension step max_step = 5, find nearby nodes radius r = 1.5, the target radius around delta_pgoal = 1, the basic RRT, original RRT * and improvement of motion path planning with RRT * algorithm respectively, as shown in figure 4.11
As shown in the three algorithms were used respectively to iterative 6000 times, 8000 times and 6000 times the path of the map.

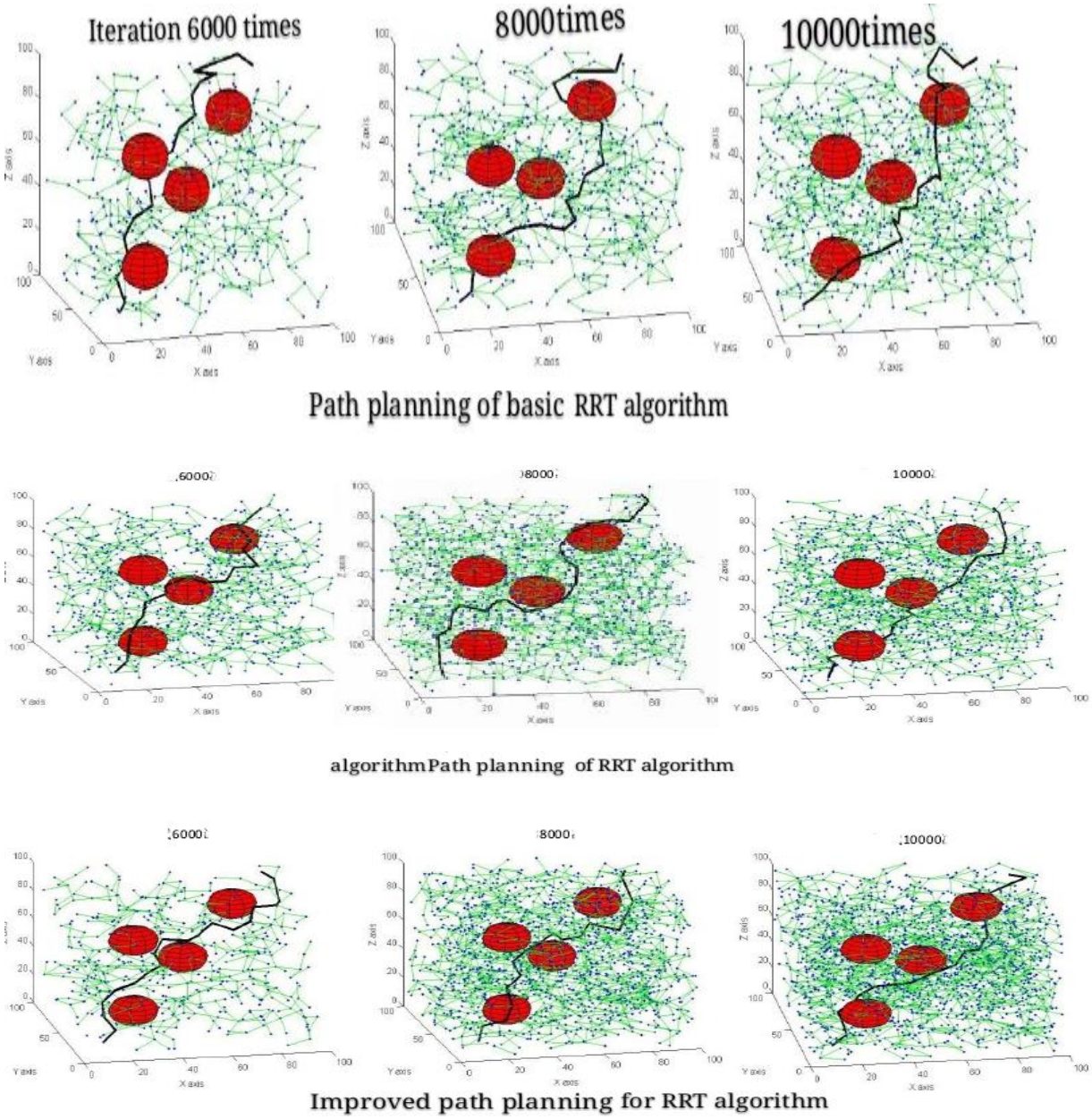


Fig. 4.3 The path planning of different algorithms in 3D space

Figure 4.3, the red is the obstacles; Blue dots for extended tree leaf nodes, on behalf of the extension process in the sampling point; Green to extension of branches dry, black represents the initial point to the target point path planning. Can be seen from the diagram, in 3 d space, planning and randomness of the path of the basic RRT algorithm[18-20], and with the increase of the number of iterations, motion path has not been any improvement; Under the same number of iterations, the improved algorithm of RRT * than the original RRT * algorithm slightly close to the optimal path, and still has gradually improved RRT * algorithm characteristics of the optimal path. Similarly, in figure 4.11 environment, select the number of iterations is 10000 times (stable state the number of iterations), the three different algorithms for 50 times independent experiment, calculate the path planning in the process of iterative search point, cost and cost Tends to the optimal path used the average of the time, as shown in table 4.2

Table 4.2 The result of three different algorithms in 3D space

algorithm	Average path planning point (one)	cost	Mean time (s)
Basic RRT	113	67	45.174
RRT algorithm	64	39.643	41.238
Improved RRT algorithm	57	24.375	14.966

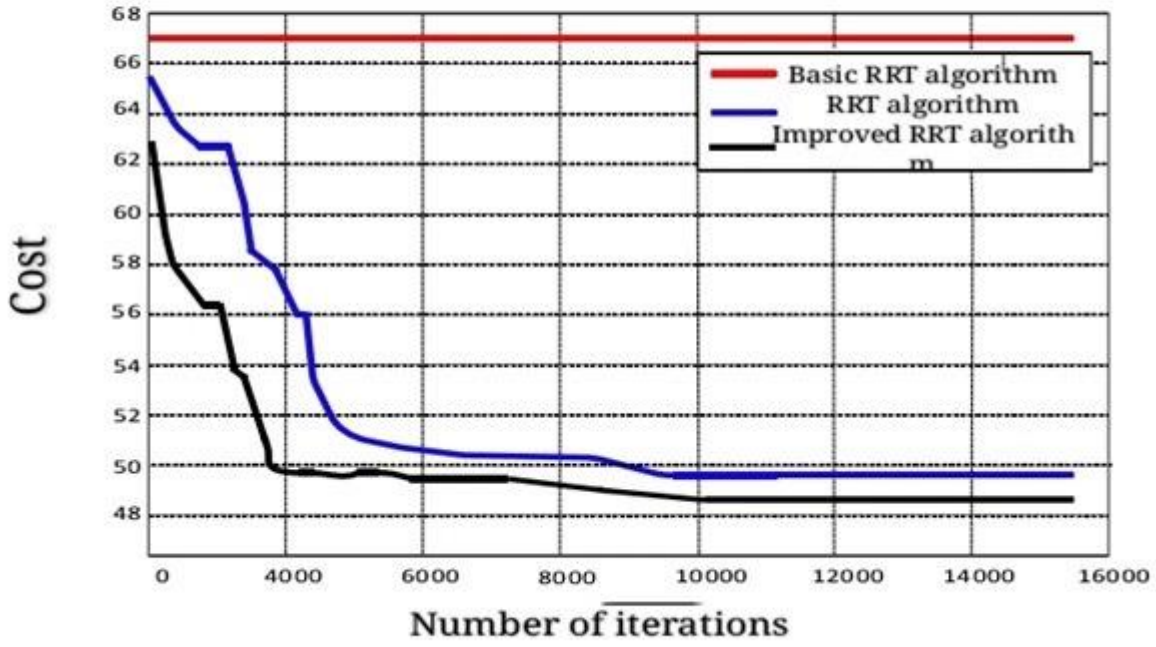


Fig. 4.4 Three different algorithms converging to the optimal path in 3D

From table 5.2, you can see that in the three-dimensional space, the improved RRT * algorithm than the basic RRT Path planning algorithm and the original RRT * average number of points is less, the average price to cost less than other two algorithms, and significantly improve the efficiency of path planning. Figure 4.12 shows along with the increase of the number of iterations, the basic RRT, original RRT * and the improved RRT * algorithm converges to the optimal path in the three-dimensional space of the trend, the same as the plane space, planning the path of the price remains the same, basic RRT algorithm RRT * algorithm and the improved RRT path gradually tend to be the most * algorithm programming Optimal, and using the improved RRT * algorithm for path planning efficiency compared with other two kinds of algorithm is improved. Comprehensive the above two kinds of environment simulation experiment, the results show that the improved RRT * algorithm not only retained the characteristics of the RRT * algorithm optimized gradually, and significantly improve the efficiency of the motion path planning.

*I. Based on the improved RRT * algorithm applied in the robot arm movement planning*

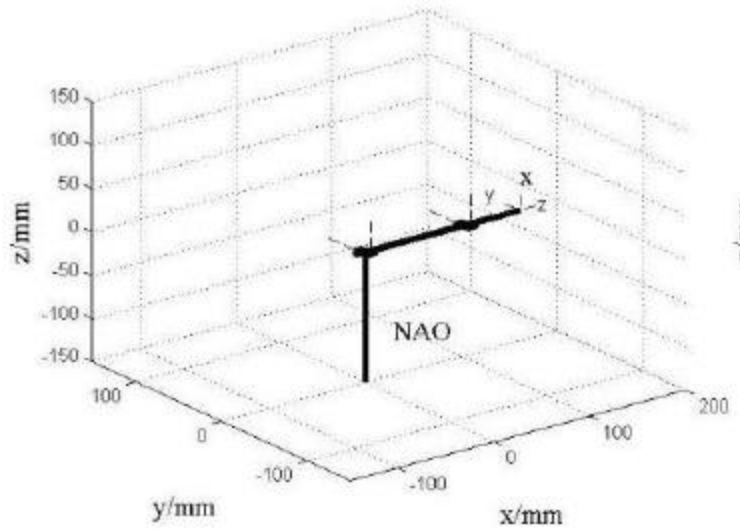
To verify the improved RRT * algorithm can exercise in high-dimensional space robot path planning, this section will be the algorithm used in humanoid robot NAO arm motion path planning, basic, at the same time RRT algorithm and original RRT * algorithm was analyzed. Because the NAO robot is a multi-degree of freedom rigid robot so according to the previous right arm NAO robot kinematics model is established. On the model by using MATLAB Robotics, offline simulation Toolbox. Selection of robot pose space for its state space because the NAO robot posture affects only the fifth joint arm transformation in the process of path planning not to consider the first joints and position of the object. To be close to the motion planning again when ready to grab the arm the process of planning the step length selection to 10 mm near nodes radius r is set to 10 mm target radius around delta_ p_{goal} = 0.5 mm

and established point set up several obstacles. Using basic RRT algorithm, RRT * algorithm and the improved RRT * algorithm 50 times independent experiments respectively and calculate the average number of path planning point on average spending and converge to the optimal path, cost and average time used, as shown in table 5.3

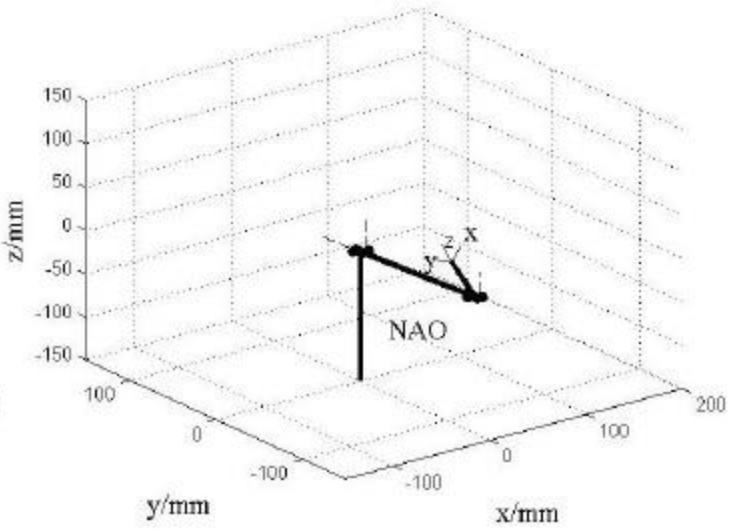
Table 4.3 The result of three different algorithms for robot arm path planning.

algorithm	Average path planning point	Average cost	Mean time(s)
RRT	68	41	29.565
RRT*	39	28.643	28.374
Improved RRT*	33	26.375	17.135

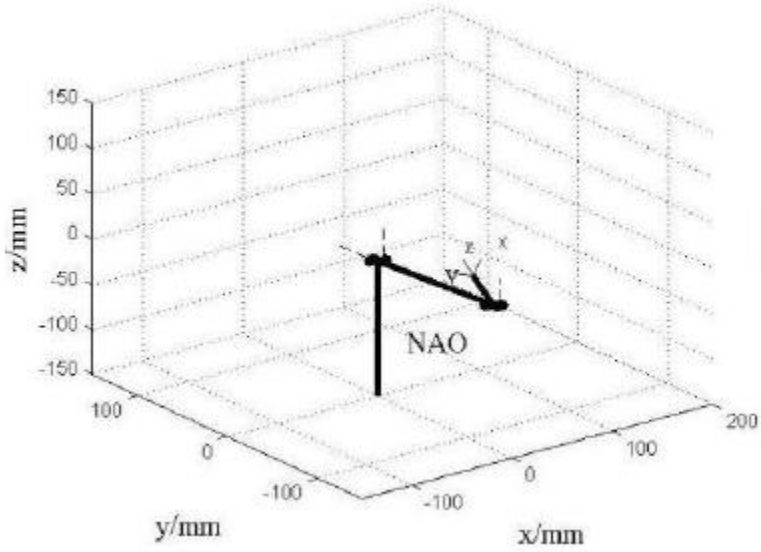
Simulation humanoid robot NAO arm using the improved RRT * algorithm. Motion path planning of the main process as shown in figure 4.5, among them, select the initial configuration of NAO robot for [0.5 °, -0.5 °, 90 °, 2 °], according to the configuration, through the robot kinematics solution of the initial state of the location of the end executor for (160 mm, 0 mm, 0), as shown in figure 4.5(a) Shown below. Figure 4.5 (b) for the robot arm close to the target object by slow mobile robot arm to target point location (45 mm, 55 mm, - - 12 mm), as shown in figure 4.5 (c), as shown in the mobile robot arm to another location as shown in figure 4.5 (d).



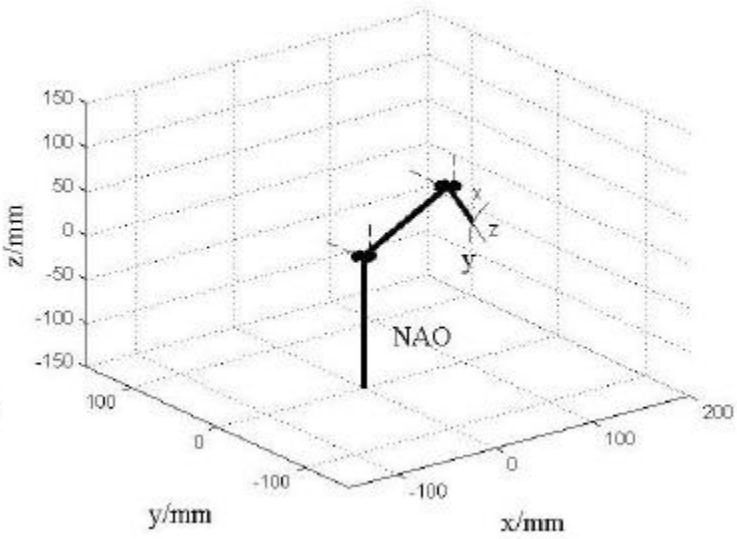
(a) the initial configuration



(b) close to the target location



(c) to arrive at the target location



(d) remove the arm

Fig. 4.5 Simulation Experiment of Robot Arm Motion Path Planning

II. No obstacle constraints object grab operation

The purpose of this experiment was to study no obstacle constraints, through the above design based on the device Object orientation, implementation of the humanoid robot with target positioning of the object, and use the change RRT * algorithm of NAO robot arm motion path planning out a feasible and finally according to the path planning by controlling a robotic arm with the target object grab operation.

NAO robot end execution for three fingers of dexterous hand, the thumb is about 2.87 cm long, the other two fingers is about 4.31 cm long, and controlled by a motor, hands up to grab 300 g items, therefore, this article selects target grab objects are moderate weight and volume of the object. first of all, through the access real-time identification of the target object and the common point is used to coordinate transformation, using coordinate transformation method, the target object coordinate transformation to the NAO robot system, thus grasping the coordinates of the point p2 (0.2276, 0.0507, 0.0507). Among them, the main process of image processing

(a) the original RGB images,
(b) the image expressed by HSV color space,
(c) shows by color threshold in HSV color space the image after segmentation,
(d) using median filter and morphological image processing

Set the NAO robot stands for the initial state, the initial point coordinate information obtained by robot position sensors, as p0, at this point p0 = (0.0627, 0.115, 0.0627), p1 to target a certain position at the upper right, his fingers when p1, and adjust the finger gestures, the set purpose is to facilitate fetching, p3, p4 to move the position of the target. One point between p1 and p2, p3 and p4 between points for the fourth chapter puts forward the improved RRT * algorithm to generate the middle path of some. Through the study of the mechanical arm of all path points inverse kinematics solution and assigned to the robot, drive robot arm by p0 movement to p1, the point through the middle path to p2, after arriving in p2 closed crawl robot fingers, then move up the object distance (convenient move objects)

P3, finally after planning out of the intermediate point movement into p4, at this point, to complete the task of the whole system operation. Figure 4.6 shows the grab motion path of success, the motion planning step length is 0.04 m.

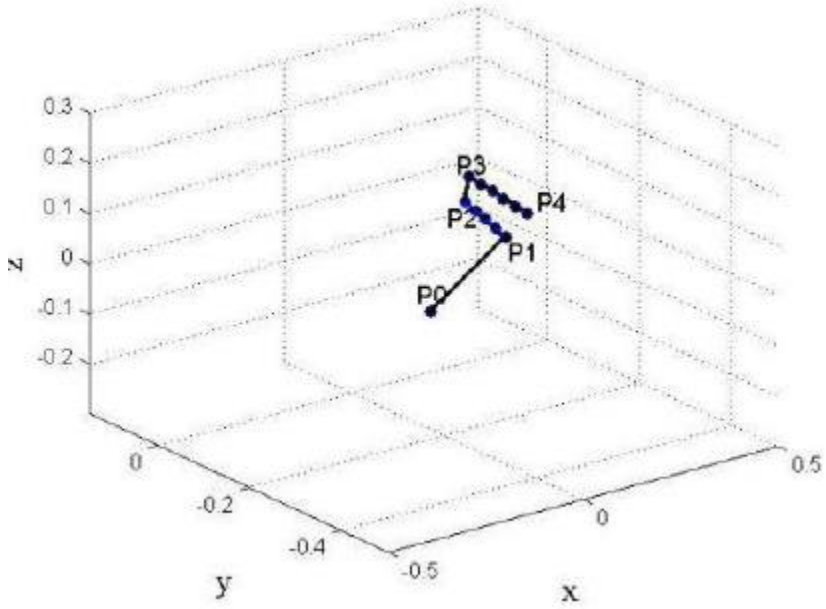


Fig. 4.6 Object grasping motion path

Can be seen from the diagram, under the condition of no obstacle constraints, arm motion path to linear form, according to the planning out of the path to the point, the control object grab robotic arm for operation.

The experimental results show that the obstacle constraints, NAO Successful completion of the robot to avoid obstacles Grab the target of the operation, verify the improvement in section in the previous chapter RRT *The reliability of the path planning algorithm.

V. *Conclusion*

As for robots and robot technology attaches great importance to both at china and abroad, and by robot is expected to release more labor, replace the tedious and repetitive work for human vision, makes the humanoid type service robot has become a hot research topic. Among them, the research on humanoid robot grab objects, the relevant domestic and foreign scholars have been in-depth study, this paper mainly studies the humanoid robot
NAO fetching operation of the object, aiming at the shortcomings of the camera is NAO robot ontology, sensors instead of robot body camera device is presented in this paper the depth perception environment, build a humanoid robot servo grab objects based on device system, has realized the robot grasping the operation of the object. In this paper, the main work is as follows:

(1) Introduces the knowledge of robot kinematics, using D-H to NAO robot arm Method and the robot kinematics equation is established the mathematical model, and by using analytical method of right arm was used to solve the inverse kinematics of robot, which laid a foundation for fetch the NAO robot objects.

(2) To achieve the target of the high precision robot objects, this paper designed a based on the device.

The humanoid robot grab target positioning system. First to image processing device for the scene information used to identify the target object and get the target detection center coordinates; And then set up the device and robot coordinate system, and constructs the bursa coordinate transformation model, and use the general linear least squares algorithm (LTLS) solve the model; According to device for center of the target position information, convert it to robot coordinate through coordinate transformation model, so as to realize the machine The person on the target object's position.

(3) To make the robots grasping objects fast, reliable, studied the robot arm motion path planning. For working space and the scope of work has clear guidance to the robot path planning, based on the analysis of the robot working space, by using iterative method and geometrical method to calculate the NAO Under the right arm robot in cartesian space of movement of scope. For NAO robot joints, the characteristics of high dimension space, using the improved RRT * algorithm for trajectory optimization, the grid search method with bidirectional search strategy is introduced into the original RRT * algorithm.

(4) For no obstacle and obstruction of constraint conditions, using it as a perceptual system, to verify the NAO robot can fetching objects of complete operation, set up different experimental environment, by combining the proposed target identification method, the application of the improved RRT * algorithm path planning for robot arm, by solving the inverse kinematics solution of the corresponding path points, and the results will be assigned to the robot, implements the fetch the robot operation. Through the above research, this article is based on To access humanoid robot servo system can realize the robot real-time grab objects, accurate and fast grab objects, mainly for the following three points：(1) on the humanoid robot NAO, based on the device object location experiment was carried out, the results show that this method has certain reliability and real-time performance, and NAO monocular vision positioning results were more accurate. (2) with the improved RRT * robotic arm motion trajectory planning algorithm, the experimental results show that this method not only retains the characteristics of RRT * gradually optimized, and the basic RRT algorithm and the programming efficiency of original RRT * algorithm is improved obviously, make the extension converges to the optimal price cost faster. (3) based on the presence of obstacles under the constraint condition of different object grab experiments, the results validate the correctness of this paper constructed the system as well as the use of the effectiveness of the method.

REFERENCES

[1] ZHENG Hongmei, YAN Yalan. Research on Positive Kinematics Analysis Method of 6R Robot[J]. Mechanical Design & Manufacture, 2014(3): 5-7.

[2] Ma Zhiyuan. Research on Arm Modeling and Control Algorithm of NAO Robot [D]. Yanshan University, 2015

[3] Wilamowski B M, Yu H. Improved computation for Levenberg–Marquardt training[J]. IEEE transactions on neural networks, 2010, 21(6): 930-937.

[4] Robotics and its intelligent control [M]. Beijing: People's Posts and Telecommunications Press, 2014.

[5] Zheng Xuelin. Research on intelligent crawling technology of humanoid robot based on visual servo[D]. Northeastern University, 2013

[6] Lysenkov I, Eruhimov V, Bradski G. Recognition and pose estimation of rigid transparent objects with a kinect sensor[J]. Robotics, 2013: 273.

[7] Zhai Yankai, Xu Yulin, Zhou Yongfei et al. Dual target setting and target localization based on plane template[J]. Computer Technology and DevelopmentExhibition, 2013, 06: 27-30+34

[8] http://doc.aldebaran.com/1-14/nao/nao-webpage.html

[9] FANG Xing, ZENG Wen-xian, LIU Jing-nan, YAO Yi-bin. General General Least Squares Algorithm for 3D Coordinate Transformation[J]. Surveying and Mapping Newspaper, 2014, 11:1139-1143.

[10] Escande A, Mansard N, Wieber P B. Hierarchical quadratic programming: Fast online

[11] humanoid-robot motion generation[J]. The International Journal of Robotics Research, 2014, 33(7): 1006-1028.

[12] Kovacic Z, Petric F, Miklic D, et al. NAO Plays a Tic-Tac-Toe Game: Intelligent Grasping and Interaction[J]. University of Zagreb, Feb, 2014.

[13] WANG Wei, LI Wei. RRT-based virtual human arm manipulation planning method[J]. Journal of System Simulation, 2009,20:6515-6518.

[14] Huang Wenbing, Sun Fuchun, Liu Huaping. Planning method considering the end motion characteristics of redundant manipulators[J]. Journal of Tsinghua University(Science and Technology) Learning Edition), 2014, 12: 1544-1548.

[15] Lozano-Perez T. Spatial planning: A configuration space approach[M]//Autonomous robot vehicles. Springer New York, 1990: 259-271.

[16] Wang Quan. RRT-based global path planning method and its application [D]. National University of Defense Technology, 2014

[17] Fontanals J, Dang-Vu B A, Porges O, et al. Integrated grasp and motion planning using independent contact regions[C]//Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on. IEEE, 2014: 887-893.

[18] Lee S J, Baek S H, Kim J H. Arm Trajectory Generation Based on RRT* for Humanoid Robot[M]//Robot Intelligence Technology and Applications 3. Springer International Publishing, 2015: 373-383.

[19] Loeve J W. Finding time-optimal trajectories for the resonating arm using the RRT* algorithm[J]. Master of Science, Faculty Mechanical, Maritime and Materials Engineering, Delft University of Technology, Delft, 2012.

[20] Adiyatov O, Varol H A. Rapidly-exploring random tree based memory efficient motion planning[C]//Mechatronics and Automation (ICMA), 2013 IEEE International Conference on. IEEE, 2013: 354-359.

AUTHORS

First Author – Posraj Khadka , ME , Nanjing Tech University, posraj_khadka@yahoo.com