2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Grasping the Object with Collision Avoidance of Wheeled Mobile Manipulator in Dynamic Environments

Pei-Wen Wu, Yu-Chi Lin, Chia-Ming Wang, and Li-Chen Fu, *Fellow, IEEE*

Abstract—**In this paper, the authors proposed a local motion planner with robot controller of the mobile manipulator to grasp the object in dynamic environments without any prior knowledge of environments. Our local motion planner is based on the concept of potential field and composed of the attractive and repulsive vectors. Then, the local motion planner decides the potential vector according to the attractive and repulsive vectors in various situations. Also, an approach to deal with the drawback of local minima is contained. The attractive and repulsive vectors are generated by the distances between the target, obstacles and the manipulator. For robot control, we take the end-effector as the control point and apply the potential vector with joint-level control, and moreover evaluate the mobility and the kinematic constraints of the robot to modify the joint velocities. The experiment platform is a wheeled mobile robot with a 5-DOF manipulator using Softkinetic DS325 which is a close range RGB-D camera as our sensor. Through several experiments, the results show that our framework is fast enough and valid to grasp the object in dynamic environments.**

I. Introduction

Nowadays, techniques in robotics are getting mature and people expect robots to do more complex tasks in human daily life. In high-level services, the ability of the robot to grasp objects is essential and the motion planning of controlling the manipulator to grasp the objects without collision has been studied for a long time. The sampling-based planning algorithms such as Rapidly-exploring trees (RRTs) [1] and Probability Roadmap (PRM) [2] algorithms are popular in recent years because of their ability to rapidly discover the connectivity of high-dimensional configuration spaces. Besides, motion planning is also considered as the trajectory optimization problem [3], [4]. By the definition of the cost function, the collision-free trajectory which is optimized with smoothness can be obtained. In above algorithms, the main research problem is to obtain a collision free path to a stationary goal. However, the goal of the manipulator during human-robot interaction is not stationary but dynamic. In this paper, to increase the robot's ability of interaction with human and environments, the authors proposed an approach for the mobile manipulator to grasp the object under dynamic environments.

The whole system is composed of three parts: (1) Perception (2) Local motion planner (3) Robot control. In the part of perception, we use color information to detect and track the target. Our local motion planner is based on the concept of potential field approach, and use the information from the RGB-D camera to calculate the attractive and repulsive vectors. Attractive vector contains the translation and rotation information to approach the target, and repulsive vector is to do collision avoidance. Then attractive and repulsive vectors are used to generate the potential vector which control the robot motion in joint-level with considering the motion of the mobile platform.

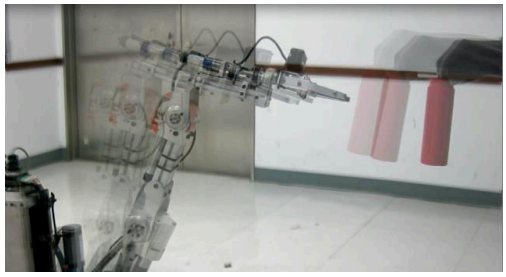


Figure 1. The robot to grasp the objects during human- robot

There are many researchers who are interested in motion planning under unknown dynamic environments [5][6][7][8][9]. Many works calculate the trajectories and represent the trajectories as the parametrized collision-free path. The path is updated as the environment changes at runtime. The elastic strip method [6] builds a collision-free "tunnel" which connects the initial and goal of the robot for avoid moving obstacles by deforming a trajectory. [7] introduces an approach which is based on PRM but considering prior-information of static and dynamic obstacles for dealing with unknown moving obstacles for mobile robots. To use virtual springs and damping elements as input, and by the concept of potential field, [8] generates the motion trajectories. [9] calculates the paths and trajectories in the *configuration* x *time* space, and updates the path by the concept of genetic algorithm.

Our method is able to grasp both static and moving objects in dynamic environments by using low cost RGB-D camera as sensor. The environment doesn't to be known or built models in advance. All we require to know is the object we want to grasp. By using depth information, we can curtail the time of computation to make our approach fast enough to adapt the moving target and environments.

The paper is organized as follows. The whole system architecture and flow of the local motion planner would be introduced in Section II. Section III are the attractive, repulsive, and potential actions which are the core of our local motion planner. Section IV shows the robot motion controller.

*Resrach supported by National Science Council of Taiwan

Pei-Wen Wu is with the Department of Electrical Engineering, NTU, Taiwan (phone: 886-2-33669885;e-mail: R00921006@ntu.edu.tw).

Yu-Chi Lin is with the Department of Electrical Engineering, NTU, Taiwan (phone: 886-2-33669885;).

Chia-Ming Wang is with the Department of Electrical Engineering, NTU, Taiwan (phone: 886-2-33669885;).

Li-Chen Fu is with the Department of Electrical Engineering and Department of Computer Science and Information Engineering, NTU, Taipei, Taiwan (phone: 886-2-23622209; fax: 886-2-23657887;
e-mail: lichen@ntu.edu.tw)

Then the experimental results are described in Section V. Finally, Section VI is the conclusions and future works.

II. System Overview

Our goal is to grasp the static or moving object, which is our target in dynamic environments. The system architecture is shown in Fig. 2. The approach in this paper uses a RGB-D camera on the wrist as sensor and the information from the encoder to calculate the kinematics of the robot, moreover, receives the external command from human to control the robot. We place the camera on the wrist and use the skill of visual servoing [10] to grasp the object. Through the local motion planner, we can obtain the potential vector, move of the base and the motion state.

The flow chart of the local motion planner is as in Fig. 3. First, we attempt to locate the target by the information from the color image. If the target exists, the attractive vector $\boldsymbol{S}_{attf}$ is going to be computed. If the target doesn't exist or after computing attractive vector, the robot computes the repulsive vector $\boldsymbol{V}_{repf}$ to do the collision avoidance. According to $\boldsymbol{S}_{attf}$ and $\boldsymbol{V}_{repf}$, the local minima escape strategy is determined to apply or not. Then, the potential vector $\boldsymbol{V}_p$, the movement of the platform $\boldsymbol{V}_{Base}$ are decided by $\boldsymbol{S}_{attf}$, $\boldsymbol{V}_{repf}$, and $\boldsymbol{P}_{ext}$. Finally, by confirming whether the target is already be grasped or not to terminate the local motion planning. If not, the robot would continue to locate the target. If yes, at this time the view of camera has been blocked by the target, so the local motion planner cannot compute $\boldsymbol{S}_{attf}$ and $\boldsymbol{V}_{repf}$ that the robot executes the command $\boldsymbol{P}_{ext}$ directly.

In the part of robot controller, $\boldsymbol{V}_p$ and $\boldsymbol{V}_{Base}$ are transferred into joint velocities $\dot{\mathbf{q}}$. $\dot{\mathbf{q}}$ are modified proportionally by the kinematic constraints to con troll the robot so that the motion of the end-effector would be safe and as anticipated.

III. Local Motion Planner

A. Attractive Action

Attractive Action is defined as how the end-effector approaches the target. The template of the target is already known, and we use color image to find the position of the target on 2D image plane. Then project the position of the target into depth space which the first two elements are the position in the plane of camera and the third is the distance to the camera. Before computing the attractive vector, we judge if the target exists or not. There are two situation that the target doesn't exist. One is that the target is really not in the view of the camera. The other is that there is no solution from the inverse kinematics with the position of the target.

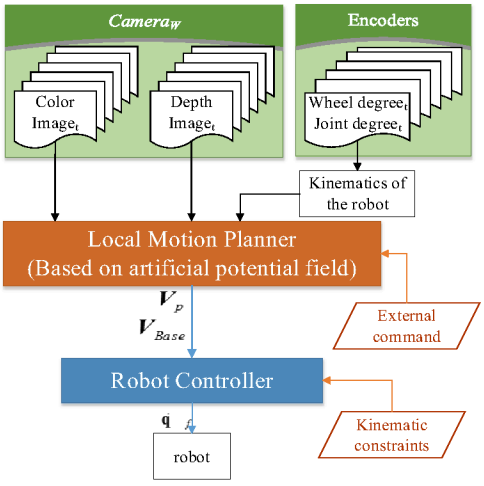


Figure. 2 System architecture.

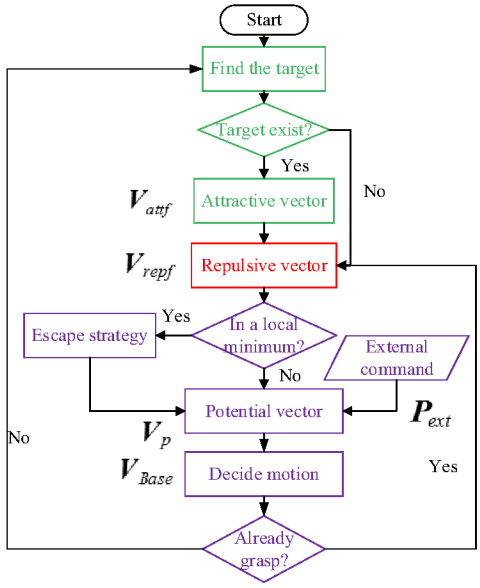


Figure. 3 Flow chart of the local motion planner

The position of the end-effector is represented in robot coordinate as $\boldsymbol{E}_R = [\,x_E^R \;\; y_E^R \;\; z_E^R \;\; \phi_E^R \;\; \theta_E^R \;\; \psi_E^R\,]^T$ which contains translation and rotation. In the perception of the RGB-D camera, we use $\boldsymbol{E}_D = [\,x_E^D \;\; y_E^D \;\; z_E^D\,]^T$ to represent translation of the end-effector and $\boldsymbol{T}_D = [\,x_T^D \;\; y_T^D \;\; z_T^D \;\; \phi_T^D \;\; \theta_T^D \;\; \psi_T^D\,]^T$ as the target. The attractive vector is defined as $\boldsymbol{S}_{att} = \left[\boldsymbol{S}_{att\_trans}{}^T \;\; \boldsymbol{S}_{att\_rot}{}^T\right]^T$, which is also composed of translation and rotation. And the translation and rotation parts relate to the position and the rotation of the target respectively.

The distance vector between the end-effector and the target is defined as

$$\boldsymbol{D}(\boldsymbol{T}_D, \boldsymbol{E}_D) = [x_T^D - x_E^D \;\; y_T^D - y_E^D \;\; z_T^D - z_E^D]^T \qquad (1)$$

We take $\boldsymbol{S}_{att\_trans} = \boldsymbol{R}_{wrist}^{-1}\boldsymbol{D}(\boldsymbol{T}_D, \boldsymbol{E}_D)$ and $\boldsymbol{S}_{att\_rot} = [\,\phi_T^D \;\; \theta_T^D \;\; \psi_T^D]^T$ directly. Because our camera is equipped on the wrist of the manipulator so that the rotation of the camera is the same as the end-effector. And ϕ_T^D, θ_T^D, and ψ_T^D are the angles of the target observed from the camera. For clarify the concept, Fig. 4 is the illustration of the attractive vector.

- *Target velocity estimation*

Our purpose is to grasp not only the static but moving object. Here we estimate the velocity by using time variation of the attractive vector, i.e., $\dot{\boldsymbol{S}}_{att} = d\boldsymbol{S}_{att}/dt$. Because the camera is on the wrist, we have to utilize $\boldsymbol{E}_R$ and $\dot{\boldsymbol{E}}_R = d\boldsymbol{E}_R/dt$ to estimate the velocity of the target. The target velocity $\boldsymbol{T}_v$ can be estimated as Eq. (2), and the final attractive vector is as Eq. (3).

$$\boldsymbol{T}_v = \dot{\boldsymbol{V}}_{att} + \dot{\boldsymbol{E}}_R \quad (2)$$

$$\boldsymbol{S}_{attf} = \boldsymbol{S}_{att} + \alpha(\boldsymbol{T}_v + \boldsymbol{V}_n) \quad (3)$$

where α is the time duration of algorithm computation, $\boldsymbol{V}_n$ is the zero mean noise. Then we transfer the attractive vector from displacement $\boldsymbol{S}_{attf}$ into velocity $\boldsymbol{V}_{attf}$ by the maximum admissible velocity. Later, the final attractive vector $\boldsymbol{V}_{attf}$ is going to decide potential action.

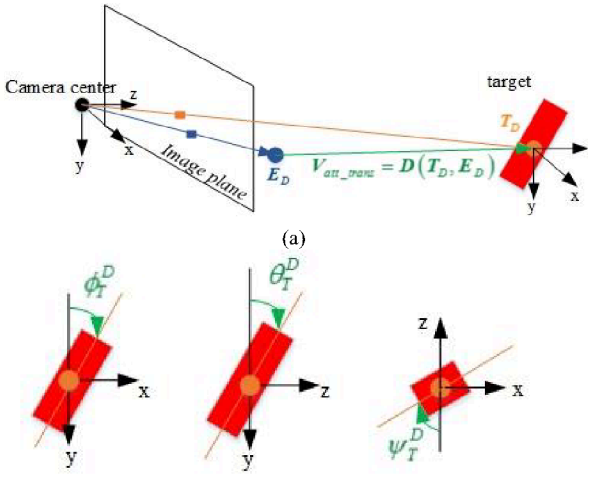


Figure. 4 The attractive vector. The red one is the target and the black axes are the camera axes. (a) shows the $\boldsymbol{E}_D$, $\boldsymbol{T}_D$, and $\boldsymbol{V}_{att\_trans}$. (b)(c)(d) show ϕ_T^D, θ_T^D, and ψ_T^D respectively.

B. *Repulsive Action*

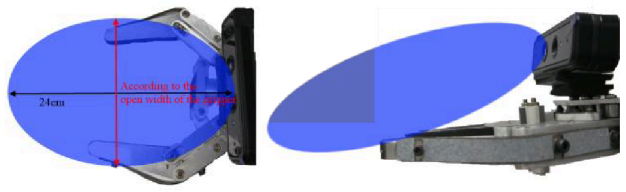
In this section, we use robot-obstacle distances to evaluate the repulsive vector to avoid collision. The collision-free space $\boldsymbol{C}$ in this paper is an ellipsoid around the end-effector as shown in Fig. 5. The minor axis of ellipse in Fig. 5(a) is adapted with the open width of the end-effector, and the major axis depends on the length of the end-effector.

The obstacle point is the point in $\boldsymbol{C}$ and represented as $\boldsymbol{O}_D=[\ x_O^D \quad y_O^D \quad z_O^D\ ]^T$ in depth space. The repulsive vector $\boldsymbol{V}_{rep} \in \Re^3$ is defined with the same direction of the distance vector $\boldsymbol{D}(\boldsymbol{E}_D, \boldsymbol{O}_D)$ as Eq. (4) with the magnitude as Eq. (5).

$$\boldsymbol{V}_{rep}(\boldsymbol{E}_D, \boldsymbol{O}_D) = \mathrm{mag}(\boldsymbol{E}_D, \boldsymbol{O}_D)\frac{\boldsymbol{D}(\boldsymbol{E}_D,\boldsymbol{O}_D)}{\|\boldsymbol{D}(\boldsymbol{E}_D,\boldsymbol{O}_D)\|} \quad (4)$$

$$\mathrm{mag}(\boldsymbol{E}_D, \boldsymbol{O}_D) = \frac{V_m}{1+\exp(\|\boldsymbol{D}(\boldsymbol{E}_D,\boldsymbol{O}_D)\|(2/\boldsymbol{C}(x_O^D, y_O^D)\text{-}1)\gamma)} \quad (5)$$

where V_m is the maximum admissible velocity, $\|\boldsymbol{D}(\boldsymbol{E}_D, \boldsymbol{O}_D)\|$ is the Euclidean distance of $\boldsymbol{D}(\boldsymbol{E}_D, \boldsymbol{O}_D)$, and γ is a shape factor. $\boldsymbol{C}(x_O^D, y_O^D)$ is the distance between $\boldsymbol{E}_D$ and the surface of collision-free space with position (x_O^D, y_O^D) in depth space. The detail to compute the repulsive vector can be found in [5].



(a) Top view (b) Side view

Figure. 5 The region of the collision-free area

- *Exclusion of the target from the collision-free space*

Sometimes the target enters the collision-free area, but the distance between the target and end-effector is still too far to grasp. At this time, the target is in the collision-free space but not obstacles. As preliminary, we've already known the width, height, and position of the target. Therefore, this information can help us to exclude the target $\boldsymbol{T}$ from $\boldsymbol{C}$ while computing the repulsive vector.

$$\mathrm{v}_{\max} = \max_{\boldsymbol{O}_D \in (\boldsymbol{C}\setminus\boldsymbol{T})} \boldsymbol{V}_{rep}(\boldsymbol{E}_D, \boldsymbol{O}_D)$$

$$\boldsymbol{V}_{rep\_t} = \mathrm{v}_{\max}\frac{\sum_{\boldsymbol{O}_D \in (\boldsymbol{C}\setminus\boldsymbol{T})} \boldsymbol{V}_{rep}(\boldsymbol{E}_D, \boldsymbol{O}_D)}{\left\|\boldsymbol{V}_{rep}(\boldsymbol{E}_D, \boldsymbol{O}_D)\right\|} \quad (6)$$

Finally, the repulsive vector is computed as Eq. (6). The direction of $\boldsymbol{V}_{rep\_t}$ relies on all obstacle points in $\boldsymbol{C}$, but the magnitude only relies on the obstacle point which is closest to the end-effector. Fig. 6 shows the difference between excluding the target from $\boldsymbol{C}$ or not. In Fig. 6(b) there are no obstacles, however the target is taken as obstacle and the repulsive vector is generated. In Fig. 6(c), the target is blocked and the judgment of the obstacle is correct; therefore, there is no repulsive vector.

- *Obstacles velocity estimation*

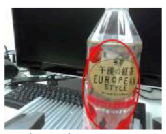
To know the velocity of moving obstacles would be helpful to enhance the action of obstacle avoidance, and we take the *Pivot Algorithm* [5] directly. Before applying the *Pivot Algorithm*, we transfer the coordinate of $\boldsymbol{V}_{rep\_t}$ to the robot coordinate, i.e., $\boldsymbol{V}_{rep\_r} = \boldsymbol{R}_{wrist}^{-1}\boldsymbol{V}_{rep\_t}$. And in *Pivot Algorithm*, the direction of $\boldsymbol{V}_{rep\_r}$ is only modified when obstacle is leaving which means that the angle between $\dot{\boldsymbol{V}}_{rep\_r}$ and $\boldsymbol{V}_{rep\_r}$ is larger than $\pi/2$, and finally we get $\boldsymbol{V}_{repf}$.

C. *Potential Action*

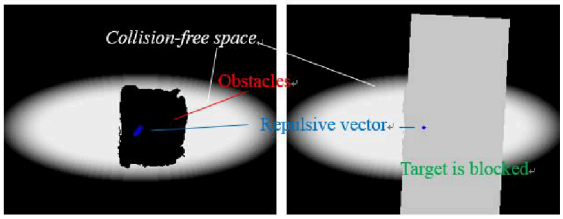
In potential action, we deal with the local minima problem which is not considered in [5], and decide the motion state of the robot.

- *Escape from the local minimum*

The drawback of artificial potential field is the local minima problem. This phenomenon occurs when there are too



(a) The target



(b) Without excluding the target (c) Excluding the target

Figure. 6 Exclusion of the target from the collision-free space

TABLE I. ADAPTATION BY ESCAPE STRATEGY

Action of $\boldsymbol{E}_R$	The repulsive vector $\boldsymbol{V}_{repf}$
ROTATE	$\boldsymbol{R}_{wrist}^{-1} * \begin{bmatrix} 0 & 0 & \phi_{gap} \end{bmatrix}^T$
UP	$\boldsymbol{R}_{wrist}^{-1} * \begin{bmatrix} 0 & 0 & \left\|\boldsymbol{D}(\boldsymbol{O}_{top}, \boldsymbol{E}_D)\right\| \end{bmatrix}^T$
RIGHT	$\boldsymbol{R}_{wrist}^{-1} * \begin{bmatrix} 0 & \left\|\boldsymbol{D}(\boldsymbol{O}_{right}, \boldsymbol{E}_D)\right\| & 0 \end{bmatrix}^T$
LEFT	$\boldsymbol{R}_{wrist}^{-1} * \begin{bmatrix} 0 & -\left\|\boldsymbol{D}(\boldsymbol{O}_{left}, \boldsymbol{E}_D)\right\| & 0 \end{bmatrix}^T$

many obstacles near the end-effector. In this paper, we utilize $\boldsymbol{V}_{repf}$ to analyze whether the local minimum occurs or not.

The situation that $\boldsymbol{V}_{repf}$ is only in z-direction and target behind obstacles implies the end-effector might in a local minimum. If the end-effector is judged might in a local minimum, there may have gaps between obstacles which the end-effector can pass through. If there are gaps between obstacles, the end-effector might be able to pass through obstacles by rotating the wrist.

First, we compute the moments of gaps by using Green's formula [11] to find the largest gap and its rotation. If the width of the gap w_{gap} is larger than the size of the end-effector, the end-effector can pass through the gap by rotating the wrist. And the repulsive vector is modified according to the rotation of the gap ϕ_{gap}. If w_{gap} is too narrow to pass or there's no gap, the manipulator would try to escape in directions of left, right, or up, by comparing $\left\|\boldsymbol{D}(\boldsymbol{O}_{left}, \boldsymbol{E}_D)\right\|$, $\left\|\boldsymbol{D}(\boldsymbol{O}_{right}, \boldsymbol{E}_D)\right\|$, and $\left\|\boldsymbol{D}(\boldsymbol{O}_{top}, \boldsymbol{E}_D)\right\|$. We don't take the direction of moving down because the static objects are usually supported by other objects such as desk.

The end-effector moves up if the $\left\|\boldsymbol{D}(\boldsymbol{O}_{top}, \boldsymbol{E}_D)\right\| < d_e$ or both $\left\|\boldsymbol{D}(\boldsymbol{O}_{left}, \boldsymbol{E}_D)\right\|$ and $\left\|\boldsymbol{D}(\boldsymbol{O}_{right}, \boldsymbol{E}_D)\right\|$ are larger than a threshold d. The setting of threshold d is to disallow the end-effector to move too much so that the target can stay in the scene, and d_e hinges on the size of $\boldsymbol{C}$. Otherwise, the end-effector would move right or left with the shorter distance. Table I. shows the adaptation of $\boldsymbol{V}_{repf}$ under different actions, and the examples of gaps exist or not are shown in Fig. 7.

- *Motion decision of the robot*

In Section II-A and II-B, the attractive and repulsive vector are already obtained. Then the potential vector $\boldsymbol{V}_p \in \Re^6$ is generated by attractive and repulsive vector, and applied to the manipulator. However, the computation of $\boldsymbol{V}_p$ is different in various cases. First of all, if the escape strategy is applied, the attractive vector would not be considered and $\boldsymbol{V}_p$ is as Eq. (7) to rotate the wrist and Eq. (8) for other three actions. (*Note*: $\boldsymbol{0} \in \Re^3$ is a zero vector.)

$$\boldsymbol{V}_p = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{V}_{repf}^T \end{bmatrix}^T \tag{7}$$

$$\boldsymbol{V}_p = \begin{bmatrix} \boldsymbol{V}_{repf}^T & \boldsymbol{0} \end{bmatrix}^T \tag{8}$$

If the target exist, $\boldsymbol{V}_{attf}$ is used directly. If the target doesn't exist i.e., $\boldsymbol{V}_{attf} = \boldsymbol{0}$, the external command $\boldsymbol{P}_{ext} \in \Re^6$ can be taken as the attraction, and $\boldsymbol{V}_{attf}$ is obtained as

$$\boldsymbol{V}_{attf} = (\boldsymbol{P}_{ext} - \boldsymbol{E}_R)/\Delta t \tag{9}$$

Then the potential vector $\boldsymbol{V}_p$ can be computed as

$$\boldsymbol{V}_p = \boldsymbol{V}_{attf} + \begin{bmatrix} \boldsymbol{V}_{repf}^T & \boldsymbol{0} \end{bmatrix}^T \tag{10}$$

After deriving the potential vector, the movement of the platform and to grasp or not are going to be decided. We set the threshold according to the workspace of the manipulator on xy-plane. Use the expected position of the end-effector and the threshold to decide the movement of the platform. Finally, $\boldsymbol{V}_{Base} \in \Re^2$ which is the linear velocity of the platform is obtained. If $\left\|\boldsymbol{V}_p\right\| < d_g$, the target is near enough to grasp. The grasping point will be found. That the center of the gripper should coincide with the position on the object which is closest to the gripper. The size of the object is smaller than the width of gripper.

IV. ROBOT MOTION CONTROL

We are going to introduce part of the robot controller in Fig. 2 in this section. To control the robot, $\boldsymbol{V}_p$ is transferred into joint velocities by pseudo Jacobian matrix as

$$\dot{\mathbf{q}} = \mathbf{J}^+(\mathbf{q})\boldsymbol{V}_p \tag{11}$$

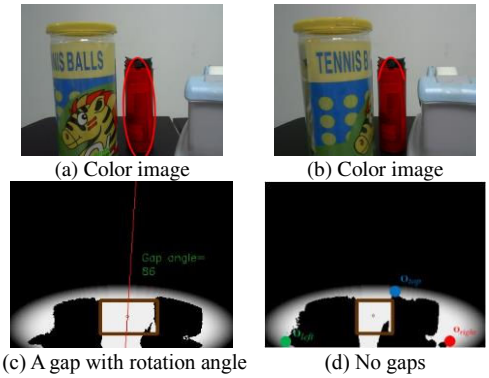
where $\dot{\mathbf{q}}$ is the velocity of each joints, $\mathbf{J}^+(\mathbf{q})$ is the pseudo Jacobian matrix.

After getting $\dot{\mathbf{q}}$, to protect the robot, we examine $\dot{\mathbf{q}}$ with the kinematic constraints. First, the joint velocities would be modified proportional according to the maximum velocity of each joint so that the movement of the robot can move as anticipate. Then we make sure the velocity won't let the joint moves out of the degree limit after executing time. Finally, we get $\dot{\mathbf{q}}_f$. After obtaining $\dot{\mathbf{q}}_f$, $\boldsymbol{V}_{Base}$ which is the linear velocity of the platform and $\dot{q}_{f,1}$ which is the angular velocity of the platform can be transferred into wheel velocity by the kinematic model of the differential wheels. Consequently, our decision, $\dot{\mathbf{q}}_f$, and the wheel velocities are applied to the control the robot.

V. EXPERIMENTS

A. Experiment set up

ARIO (Agile Robot In Office) is our mobile robot composed of differential wheels and a 5 DOF redundant manipulator. It's equipped with Softkinetic DS325[12] on wrist as in Fig. 8. Softkinetic DS325 is a near range RGB-D camera whose efficient distance is between 0.15 to 1 meters.



(a) Color image (b) Color image

(c) A gap with rotation angle (d) No gaps

Figure. 7 Examples of gaps exist or not. The red bottle is our target (a)(b) are the color images. The first column is the situation that the end-effector can pass a gap. The second column is the situation requires to calculate the distance in three directions.

Figure. 8 Photo of ARIO with DS325

The resolution of depth image is 320*240 pixels and 640*480 pixels of the color image.

In our experiment, the tracking part is implemented by Camshift [13] with color information. Furthermore, because the mechanism of our manipulator lacks degree of freedom, ψ_T^D is set as zero, and the escape strategy is simplified to rotate or move up. The local motion planner re-plans the path about every 0.25 seconds, and the robot executes each motions about every 0.3 seconds. The other parameters are set as follows in following experiments.

$\boldsymbol{E}_D = [160 \quad 120 \quad 0]^T, V_m = 25$ cm/s, γ=4, $d_g = 20$ cm.

Results

We have three experiments to verify our approach. The first is to grasp the moving object. Second is to grasp the object with dynamic obstacles. And the third is to show our escape strategy.

- Grasping the moving target

In this experiment, the ability to grasp the moving target is tested. The experiment process is as in Fig. 9. At beginning, the target is at right side of the end-effector, so the robot turns right. In Fig. 9(a)-(d), the target is moving, and robot attempts to grasp the target. In Fig. 9(e)-(h), the target is rotated and moving, and the end-effector is rotating and moving with the target. Finally, the robot grasps the target.

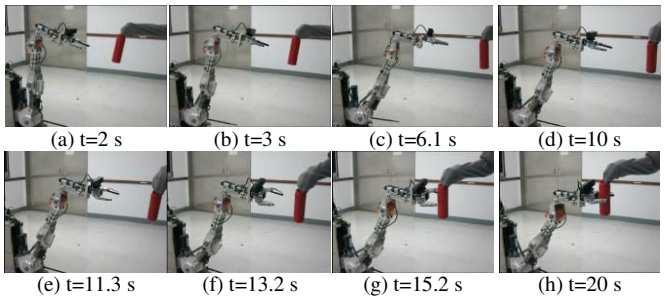
The trajectories of the end-effector and the target are shown in Fig. 10. Two trajectories don't meet at the end because the distance is too close to draw the correct trajectory of the target. From the trajectories, the movements of the end-effector and target can be seen distinctly that the end-effector aims to reach the position of the target. From t=6s to 12s, the target is out of the present workspace and the platform is moving forward. For not shock the human, we set the velocity very slow, so the effect might not so notable to see.

- Grasping the target with collision avoidance

In this experiment, we show the ability of the robot to avoid collision and to grasp the object. The target is the red bottle on the desk, the cup is the obstacle, and the process of grasping is shown in Fig. 11.

In Fig. 11(b)(c), the obstacle appears and the end-effector avoid the collision successfully. In Fig. 11(d)(e), the target is moved, and robot moves to the target. Finally, the robot grasps the target in Fig. 11(f).

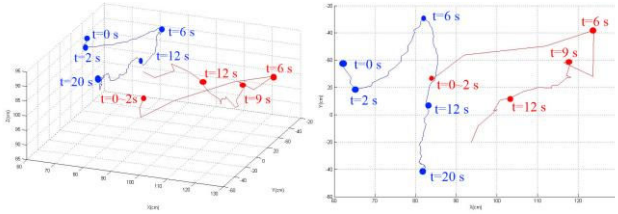
Fig. 12 shows the trajectory of the end-effector with potential vector in robot coordinate. The end-effector is doing collision avoidance from t=4 to 8 s to avoid the arm of human and the cup. Other time it's moving toward to the target even the target is moved.



(a) t=2 s (b) t=3 s (c) t=6.1 s (d) t=10 s

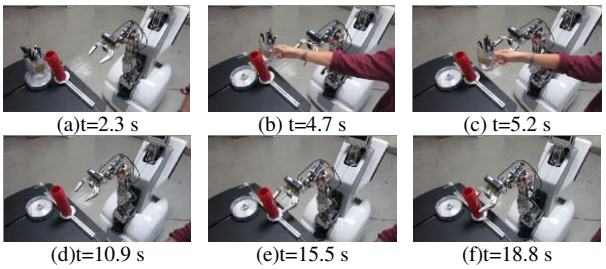
(e) t=11.3 s (f) t=13.2 s (g) t=15.2 s (h) t=20 s

Figure. 9 Grasping the moving target



(a) 45 angle view (b) Top view

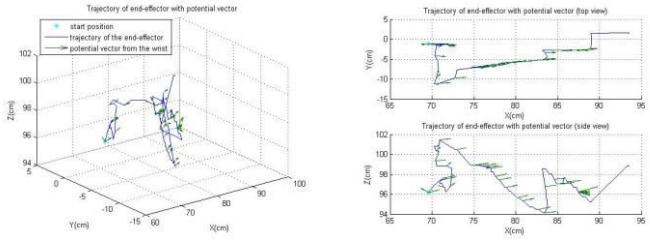
Figure. 10 The trajectory in experiment I. The red line is the trajectory of the target. The blue line is the trajectory of the end-effector.



(a)t=2.3 s (b) t=4.7 s (c) t=5.2 s

(d)t=10.9 s (e)t=15.5 s (f)t=18.8 s

Figure. 11 Grasping the target with collision avoidance.



(a) 45 angle view (b) Top and side view

Figure. 12 The trajectory in experiment II. The green vector is the potential vector. The blue line is the trajectory of the end-effector.

- To escape a local minimum

The experiments of escape local minimum are to test our escape strategy. As we described before, we only apply gap exist or moving up the end-effector two strategies to our robot. Fig. 13 is the image sequences of the robot motion that find the gap which the gripper can pass directly. And Fig. 14 is the trajectory with the potential vector in this experiment. Fig. 15 is the process that the end-effector keeps to move up and forward, finally the target is grasped. And Fig. 16 is the trajectory with the potential vector in this experiment. The

trajectory of the end-effector goes up and down the same as the image sequence in Fig. 15.

From above results, our planner and robot controller are reliable and verified to grasp the target under different cases. However, we face some issues during experiments. The reasons that we fail to grasp the object. One is the tracking error, and another is that the target moves out of the view of the camera. Other issues we met most relate to the hardware and physical constraints of our manipulator, such as the closing speed of the gripper that we have to wait the gripper closing when grasping the moving object.

VI. Conclusions

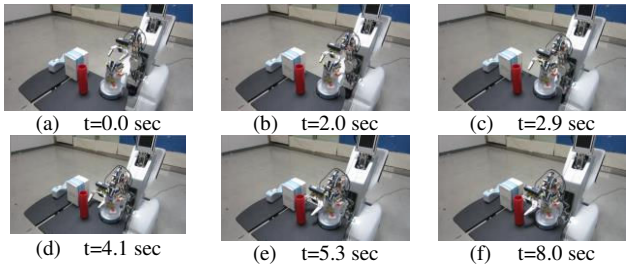
In this paper, authors proposed a framework of local motion planner with robot controller which can grasp the object in dynamic environments. In our method, we don't require any prior information of the environment. The attractive and repulsive actions are derived by the RGB-D camera. Furthermore, according the motion decision, the potential vector is generated to control the robot and can deal with the problem of local minima. The experimental results verify that our system is feasible and the local motion planner is valid to grasp the target and avoid dynamic obstacles.

About the future works, the main objective is to consider the collision avoidance of the whole manipulator and the case of lose target by using another camera on the robot to sense global information.

(http://youtu.be/PStJlAvp5iA)

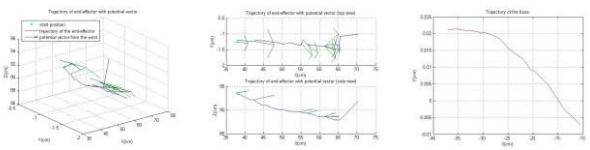
Acknowledgment

This work was supported by the National Science Council, R.O.C. under Grant NSC-100-2221-E-002-096.



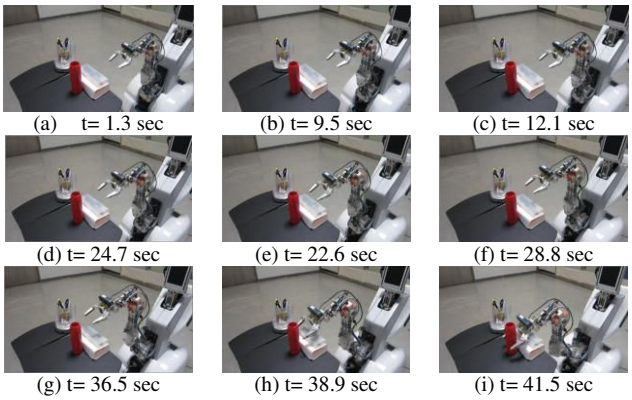
(a) t=0.0 sec (b) t=2.0 sec (c) t=2.9 sec (d) t=4.1 sec (e) t=5.3 sec (f) t=8.0 sec

Figure. 13 Local minima escape strategy (*GAP*)



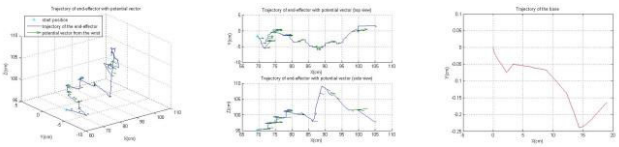
(a) 45 angle view (b) Top and side view (c) The base

Figure. 14 The trajectory of the robot (*GAP*)



(a) t= 1.3 sec (b) t= 9.5 sec (c) t= 12.1 sec (d) t= 24.7 sec (e) t= 22.6 sec (f) t= 28.8 sec (g) t= 36.5 sec (h) t= 38.9 sec (i) t= 41.5 sec

Figure. 15 Local minima escape strategy (*UP*)



(a) 45 angle view (b) Top and side view (c) The base

Figure. 16 The trajectory of the robot (*UP*)

References

[1] J.J. Kuffner, Jr. and S.M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. 2000 IEEE Int. Conf. Robotics and Automation,* pp. 995-1001.

[2] L. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration space," *IEEE Trans. Robotics and Automation*, vol. 12, no.4, pp.566–580, 1996.

[3] N. Ratliff, M. Zucker, J. A. Bagnell and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE Int. Conf. Robotics and Automation*, pp. 489-494.

[4] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *2011 IEEE Int. Conf. Robotics and Automation*, pp. 4569-4574.

[5] F. Flacco, T. Kroger, A. De Luca and O. Khatib. "A depth space approach to human-robot collision avoidance," in *2012 IEEE Int. Conf. Robotics and Automation*, pp. 338-345.

[6] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *Int. J. of Robotics Research*, vol. 21, no. 12, pp.1031-1052, 2002.

[7] J. van den Berg, D. Ferguson, and J. Kuffner, "Anytime path planning and replanning in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May, 2006, pp. 1243–1248.

[8] S. Haddadin, H. Urbanek, S. Parusel, D. Burschka, J. Rossmann, A. Albu-Schaffer and G. Hirzinger, "Real-time reactive motion generation based on variable attractor dynamics and shaped velocities," in *2010 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 3109-3116.

[9] J. Vannoy, and X. Jing, "Real-time adaptive motion planning (RAMP) of mobile manipulators in dynamic environments with unforeseen changes," *IEEE Trans. Robotics,* vol. 24, no. 5, pp. 1199-1212, 2008.

[10] F. Chaumette, S. Hutchinson, "Visual servo control, Part I: Basic approaches," in *IEEE Robotics and Automation Magazine*, pp. 82-90, 2006

[11] W. Kaplan, "Green's theorem." Section 5.5 in *Advanced Calculus,* 4th ed, Reading. MA: Addison-Wesley, 1991, pp. 286-291.

[12] Softkinematic http://www.softkinetic.com/

[13] G.R. Bradski, "Computer vision face tracking for use in a perceptual user Interface", Intel, 1998.