

Vision-Based Cooperative Manipulation of Mobile Robots

Wen-Chung Chang and Jing-Yan Ju-Ge
 Department of Electrical Engineering
 National Taipei University of Technology
 1 Sec 3 Chuna-Hsiao E Rd, Taipei 106, Taiwan
 E-mail: wchang@ntut.edu.tw

Abstract

A robot cooperative manipulation system based on visual simultaneous localization and mapping is proposed in this paper. A single camera mounted on each mobile robot is used to observe unknown environments. SURF algorithm is employed to extract features of the environment and the target object. Three-dimensional positions of the features are estimated according to the visual information. The states of the robot and matched features are updated using extended Kalman filtering to reconstruct a consistent environment map based on which path planning and robot navigation are further accomplished. To carry out cooperative manipulation tasks, a compliance control approach is adopted to avoid dropping the object and ensure completing manipulation tasks successfully.

1 Introduction

Automated guided vehicles have been widely utilized in industrial environments. Those vehicles move around by sensing wires buried under the floor or reflective paints coated on the floor. Therefore, their actions are lack of autonomy and the fixed motion path must be planned in advance. Existing robotic transportation can be performed by pushing or pulling the object [1]. Although easier to control, they are not reliable because the object will experience frictional force against the ground. When the gravity center of the object is too high, the object is likely to be dumped and damaged. To resolve this problem, Wang *et al.* [2] utilized a 1-DOF manipulator to complete the multi-robot cooperative manipulation and transportation task. Hirata *et al.* [3] considered the force of the operator to complete the cooperative manipulation and transportation task with robots. Chang *et al.* [4] combined the visual intelligent space with the information measured by the onboard cameras to design a visual servo control approach to complete multi-robot cooperative manipulation and transportation tasks. To resolve drawbacks in existing approaches, a multi-robot cooperative manipulation approach is integrated with high autonomy, flexibility, and reliability. In particular, the proposed system employs onboard vision for localization and mapping that can be further integrated to accomplish cooperative manipulation tasks successfully.

2 System Description

A robot cooperative manipulation system based on visual simultaneous localization and mapping is proposed in this paper. The system is composed of mobile robots with onboard dual-arm manipulators, cam-

era and motor sensors, wireless communication units, and personal computers. The system configuration is shown in Fig. 1. A single camera mounted on the mobile robot is used to capture environment information sending to a personal computer via wireless networking. Real-time images are processed and control commands are computed and then sent to the mobile robots through the wireless network. When executing the manipulation and transportation task, the motor sensors mounted on each joint of the dual-arm manipulator provide required information for the purpose of implementing effective compliance control.

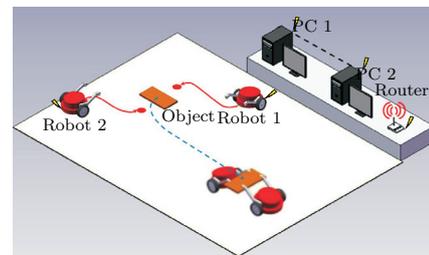


Figure 1. System configuration.

Once the system is started, the SLAM algorithm is initiated. The single camera mounted on each robot is used to capture environment images to be processed for SURF features. Therefore, data association can be established by matching SURF features in subsequent images. The SLAM results can thus be effectively updated. Therefore, accurate robot position and a consistent map can be made possible. Based on the updated map, the mobile robots can be navigated to follow a planned trajectory and perform object manipulation and transportation tasks. When the object is located, two mobile robots are controlled to grab the object simultaneously. Cooperative manipulation and transportation task can then be started. One robot is considered as the follower, and its motion must comply with the motion of the leader robot. According to each joint motor state, the torque command for each joint can be determined for the compliance motion of the arm.

2.1 Feature detection and matching

SURF (Speeded Up Robust Features) algorithm [5] based on Hessian matrix is utilized to detect and describe the environment features. Hessian matrix is defined as follows:

$$\mathbf{H}(x, y) = \begin{bmatrix} L_{xx}(x, y) & L_{xy}(x, y) \\ L_{xy}(x, y) & L_{yy}(x, y) \end{bmatrix} \quad (1)$$

After the approximation mask is used, the convolution results L_{xx} , L_{yy} , L_{xy} are substituted for D_{xx} , D_{yy} , D_{xy} . The determinant of Hessian matrix can be approximated as

$$\det(\mathbf{H}_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad (2)$$

2.2 Modeling and estimation

In this paper, extended Kalman filtering is utilized to estimate the probability distribution of robot pose. The onboard visual sensor is used to detect SURF features. Feature matching in the sequence of real-time images is performed for data association. These results are utilized to update robot pose in the world frame. In this paper, the inverse depth parametrization method [6] is adopted to estimate the depth of features by monocular vision. The uncertainty of the robot pose estimate can be computed by the motion model. The camera state is regarded as the robot state as follows.

$$\mathbf{x}_C = [\mathbf{r}^T \quad \phi^T \quad \nu^T \quad \omega^T]^T \quad (3)$$

where \mathbf{r} represents the camera position, ϕ represents the camera orientation, ν is the linear velocity of the camera, and ω is the angular velocity of the camera. The encoders mounted on the left and right wheels of the mobile robot are used to estimate the robot pose roughly. The estimation error is due to the instrument specifications, uneven ground, or wheel slip. In the motion model, the estimation error is treated as the camera velocity noise as follows.

$$\mathbf{w} = \begin{bmatrix} w_\nu \\ w_\omega \end{bmatrix} = \begin{bmatrix} a\Delta t \\ \alpha\Delta t \end{bmatrix} \quad (4)$$

where a and α is the camera linear and angular acceleration caused by the camera velocity noise, respectively, Δt is the sampling time, and w_ν and w_ω are the camera velocity noise. Therefore, the motion model is defined as

$$\mathbf{x}_{C_k} = f(\mathbf{x}_{C_{k-1}}, \mathbf{w}_{k-1}) \quad (5)$$

where $f(\cdot)$ represents the state transition function.

The uncertainty of the sensor measurements can be computed by the measurement model. In this research, a single camera is used to observe the environment, and the measurement vector is defined as

$$\mathbf{z}_k = [\mathbf{z}_{1_k}^T \quad \mathbf{z}_{2_k}^T \quad \cdots \quad \mathbf{z}_{m_k}^T]^T \quad (6)$$

where m is the number of features captured by the camera at time k and \mathbf{z}_{i_k} is the coordinates of feature i in the current image frame. The feature coordinates are obtained directly from image information. The perspective projection model is adopted as the measurement model, which is defined as

$$\hat{\mathbf{z}}_{i_k} = g(\mathbf{x}_k, \mathbf{n}_k) = \begin{bmatrix} u_0 + f \frac{c h_{x_i}}{e h_{z_i}} \\ v_0 + f \frac{c h_{y_i}}{e h_{z_i}} \end{bmatrix} \quad (7)$$

where $g(\cdot)$ represents the measurement function and \mathbf{n} is the measurement noise. The reason why the measurement noise exists could be due to different environment conditions such as temperature, humidity, and

reflectivity. Furthermore, (u_0, v_0) represents the center of image plane, f represents the focal lens, and ${}^C h_{x_i}$, ${}^C h_{y_i}$ and ${}^C h_{z_i}$ are components of the ray vector, which are the feature coordinates with respect to the camera frame.

In this research, a single camera is used to observe the environment. However, the distance between the camera and the object can not be measured in real time. Thus, the inverse depth parametrization method is adopted to solve the problem, and the three-dimension position is represented by the following six-dimension state vector.

$$\mathbf{m}_i = [r_{x_i} \quad r_{y_i} \quad r_{z_i} \quad {}^W\theta_i \quad {}^W\psi_i \quad \rho_i]^T \quad (8)$$

where r_{x_i} , r_{y_i} and r_{z_i} are the camera position when the feature i is detected for the first time, ${}^W\theta_i$ represents the camera azimuth with respect to the world frame, that is the angle between the projection of the ray vector in the x-z plane and the x-axis, ${}^W\psi_i$ represents the camera elevation with respect to the world frame, that is the angle between the projection of the ray vector in the x-z plane and the ray vector, and ρ_i represents the inverse depth. When a new feature is detected, a six-dimension state vector is generated to be added to the EKF state. Furthermore, the inverse depth is predicted and updated by EKF until the value converges. The azimuth and elevation is calculated as

$$\begin{bmatrix} {}^W\theta_i \\ {}^W\psi_i \end{bmatrix} = \begin{bmatrix} \arctan(\frac{{}^W h_{z_i}}{{}^W h_{x_i}}) \\ \arctan(\frac{{}^W h_{y_i}}{\sqrt{({}^W h_{x_i})^2 + ({}^W h_{z_i})^2}}) \end{bmatrix}. \quad (9)$$

The six-dimension state can be converted back to the three-dimension position with respect to the world frame by

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \begin{bmatrix} r_{x_i} \\ r_{y_i} \\ r_{z_i} \end{bmatrix} + \frac{1}{\rho_i} \begin{bmatrix} c({}^W\theta_i)c({}^W\psi_i) \\ s({}^W\psi_i) \\ s({}^W\theta_i)c({}^W\psi_i) \end{bmatrix}. \quad (10)$$

3 Path Planning and Navigation

Based on the positions of the robot, the target, and the obstacles, a heuristic method can be developed to perform path planning for the mobile robots. In Fig. 2(a), the grid of the robot location is considered as the start grid, which is denoted as the blue grid. The red grid indicates the target grid. The black grids indicate the obstacles. Take the start grid as the current grid, and add it to the closed list that we do not need to check it again. The adjacent grids of the current grid are added to the open list that we need to check, and the current grid is considered as their parent grid. In the figure, each grid has a pointer that points back to its parent grid. Then, the moving cost F, G and H of each grid are determined as follows. Specifically, the numbers within each grid from top to bottom is the move cost G, H, and F, respectively. G is the moving cost from the start grid to the considered grid, following the path generated to get there. Each horizontal or vertical move is assigned a cost of 10, and each diagonal move is assigned a cost of 14. H is the estimated moving cost to move from the considered grid to the target grid. H can be estimated in a variety of ways. The method we adopt here is the

Manhattan method, where the total number of grids moved horizontally and vertically to reach the target grid from the current grid is calculated, and then multiply the total number by 10. The diagonal move and any obstacles that may be in the path are ignored, thus the value of H is an estimation for the remaining distance, not the actual value. This is indeed a heuristic method. Furthermore, the sum of move cost is defined as $F = G + H$.

To continue the search, the lowest F score grid from all those that are on the open list is chosen to do the following steps. Drop the chosen grid from the open list and add it to the closed list, and consider the chosen grid as the current grid. Check all of the adjacent grids of the current grid, and then three different conditions need to be considered. Firstly, if the grid can not pass or it has been on the closed list, ignore it. Secondly, if the grid is not on the open list, then add it to the open list. Moreover, the current grid is considered as its parent grid, and calculate its F, G and H. Thirdly, if the grid has been on the open list, then G score is used to check if the new path is better or not. If it is, then the current grid is considered as the parent grid for this grid, and recalculate the moving cost F, G and H for this grid. Refer to Fig. 2(b) to 2(e), we can progressively observe the operation of the above steps. The case when the target grid is added to the closed list is shown in Fig. 2(e), which indicates that the path is found and the above steps can now be terminated. If failed to find the target grid and the open list is empty, it appears that no path can be determined. Finally, working backwards from the target grid, go from each grid to its parent grid until the start grid is reached to generate the determined path, which is shown in Fig. 2(f).

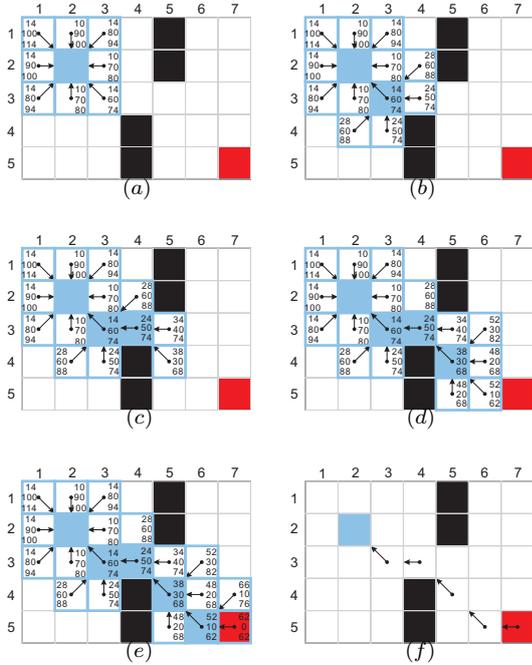


Figure 2. Path planning.

In order to obtain a smooth path for the mobile robot, the grid centers along the determined path are

used to generate the Bezier curve as follows.

$$\Gamma(t) = \sum_{i=0}^n p_i \frac{n!}{i!(n-i)!} (1-t)^{n-i} t^i \quad (11)$$

The formula can be expanded as

$$\Gamma(t) = p_0(1-t)^n + \binom{n}{1} p_1(1-t)^{n-1}t + \dots + p_n t^n \quad (12)$$

where p_i denotes the grid centers along the path, $t \in [0, 1]$.

In order to drive the mobile robot to move along the planned path, a fixed gain controller is proposed. The encoded errors are defined as

$$\begin{cases} e_\phi = \phi_d - \phi_z \\ e_d = \sqrt{(x_d - r_x)^2 + (y_d - r_y)^2} \end{cases} \quad (13)$$

The fixed gain controller is proposed as follows.

$$\begin{cases} \omega_L = K_d e_d + K_\phi e_\phi \\ \omega_R = K_d e_d - K_\phi e_\phi \end{cases} \quad (14)$$

where ω_L and ω_R represent the angular velocities of the left and right wheels, respectively.

4 Cooperative Manipulation

4.1 Grasping task

In order to implement the robot cooperative manipulation and transportation task, the mobile robot is controlled to move towards the object, and then control the manipulator to grab the object. Two features are attached to the object, and their SURF descriptions are determined a priori. Therefore, when the mobile robot explores the environment, the system tries to match them and reconstruct their position with respect to the world frame. The reconstruction results are then be used to guide the manipulator to grab the object. In particular, desired joint positions are determined by the inverse kinematics. When the relative pose between the mobile robot and the object reaches the desired one, the manipulator is driven to grab the object. The position of the end effector with respect to the world frame can be determined as follows.

$$\begin{bmatrix} W_x \\ W_y \\ W_z \\ 1 \end{bmatrix} = {}^C T_0^C T_6^0 T \begin{bmatrix} 6_x \\ 6_y \\ 6_z \\ 1 \end{bmatrix} \quad (15)$$

The end effector of the dual-arm mobile robot can thus be controlled to reach the object. Upon the object is reached, the gripper is closed to complete the grab task.

4.2 Compliance control

In order to avoid dropping the object caused by making turns when the robot cooperative manipulation and transportation task executes, a compliance control approach is introduced to ensure completing the task successfully. The manipulator can be modelled as follows.

$$\tau = \mathbf{J}^T(\Theta) K_{px} \mathbf{J}(\Theta) \delta \Theta \quad (16)$$

Therefore, the position controller is designed as

$$\tau_p = \mathbf{J}^T(\Theta)K_{px}\mathbf{J}(\Theta)\mathbf{E}_\Theta + K_v\dot{\mathbf{E}}_\Theta \quad (17)$$

where $\mathbf{E}_\Theta = \Theta_d - \Theta$ and Θ_d are calculated as follows. Firstly, the angular velocities of the left and the right wheels are returned from the encoders mounted on the wheels. Their difference is considered to determine the elongation required for the manipulator. In this research, the difference between the left and the right wheel velocities is multiplied by 0.5 to obtain the elongation in practice. When the manipulator is controlled to reach the elongation, the original height and pose of the end effector must be maintained. The desired position of the end effector X_d is determined based on above rules, which is used to further calculate the desired position of each joint Θ_d based on the inverse kinematics. The force error of the end effector is defined as $\mathbf{E}_F = \mathbf{F}_d - \mathbf{F}$, where the measured force can be computed as $\mathbf{F} = \mathbf{J}^{-T}(\Theta)\tau$. The force controller is thus designed as

$$\tau_f = K_p\mathbf{J}^T(\Theta)\mathbf{E}_F. \quad (18)$$

Furthermore, the proposed hybrid position and force controller is as follows.

$$\tau = \tau_p + \tau_f. \quad (19)$$

5 Experiments

The proposed vision-based cooperative manipulation approach was effectively implemented in a laboratory environment with a leader-follower manipulation task. According to the real-time updated maps, the leader and follower robots move towards the target object until the grasping task is completed. Then the two robots transport the object back to the initial position of the leader robot. The manipulation scene when the mobile robots execute the cooperative manipulation and transportation task is illustrated in Fig. 3. The corresponding trajectories of the leader and the follower robots can be seen in the left and the right of Fig. 4, respectively. The blue, red, and green lines indicate the trajectories of the robot, the left arm, and the right arm, respectively. Based on the results, the left arm of the leader robot and the right arm of the follower robot appear to stretch out due to compliance control when making the turn. One can also observe unsmooth controlled trajectories that are clearly caused by the uneven ground.

6 Conclusion

In the paper, SURF algorithm is employed to detect environment features with a single onboard camera. Matched features in the real-time observed images are used to establish correct data association thus ensuring consistent mapping to be updated through iterating EKF-SLAM algorithm. According to the updated map, a path planning method and a compliance controller are employed for performing the cooperative manipulation task. According to the experiment results, the proposed system appears to be capable of performing cooperative manipulation tasks effectively in a real laboratory environment.

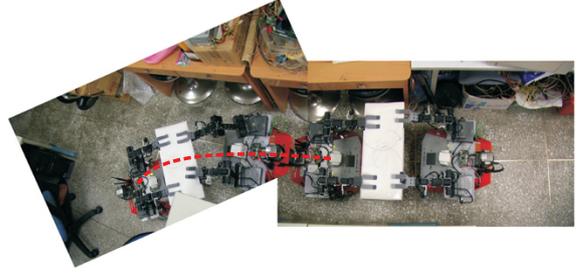


Figure 3. Manipulation scene where the leader {left} is moving backward towards the left with a turn with the follower.

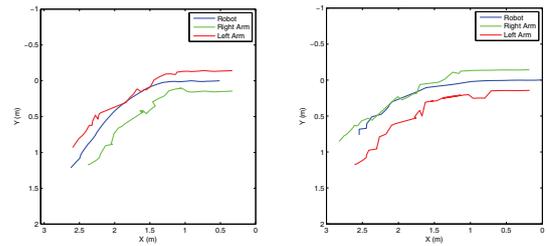


Figure 4. Trajectories of the leader {left} and the follower {right} robots.

Acknowledgments

This research was supported by the National Science Council of Taiwan, Republic of China under grant NSC 101-2221-E-027-058-MY3.

References

- [1] A. Yamashita, J. Sasaki, J. Ota, and T. Arai. Cooperative manipulation of objects by multiple mobile robots with tools. In *Proc. of the 4th Japan-France/2nd Asia-Europe Congress on Mechatronics*, pages 310–315, 1998.
- [2] Z. D. Wang, E. Nakano, and T. Takahashi. Solving function distribution and behavior design problem for cooperative object handling by multiple mobile robots. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 33(5):537–549, Sep. 2003.
- [3] Y. Hirata, Y. Ojima, and K. Kosuge. Distributed motion control of multiple passive object handling robots considering feasible region of brake control. In *Proc. of the 2010 IEEE International Conference on Robotics and Automation*, pages 3413–3419, Anchorage, Alaska, USA, May 2010.
- [4] W.-C. Chang and V.-T. Nguyen. Control of cooperative dual-arm mobile robots in a vision-based intelligent space. In *Proc. of the 13th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, Nagoya, Japan, Sep. 2010.
- [5] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110:346–359, 2008.
- [6] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse depth parametrization for monocular slam. *IEEE Transactions on Robotics*, 24(5):932–945, Oct. 2008.