2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Robot Learning Based on Partial Observable Markov Decision Process in Unstructured Environment

Hongtai Cheng, *Member, IEEE,* Heping Chen, *Senior Member, IEEE,* Lina Hao and Wei Li

Abstract— **Robot teaching is necessary for the current industrial robot applications. Because work stations have to be stopped to perform teaching processes, the manufacturing efficiency is decreased. In this paper we propose to utilize an uncalibrated vision system mounted on a mobile robot ("Adult" robot) with learning capability to supervise a group of fixed robots ("Child" robots) to accomplish a robot teaching task automatically without stopping work stations. To increase the system flexibility, hand-eye calibration and calibration between the robots are eliminated. A Partial Observable Markov Decision Process(POMDP) is formulated and solved using the Successive Approximation of the Reachable Space under Optimal Policies (SARSOP) algorithm to enable the teaching process using image features with uncertainties. The proposed algorithm was tested using the "adult" robot to teach a "child" robot to perform a high accuracy peg-in-hole assembly process. The experimental results verify the effectiveness of the proposed approach. The proposed method can also be used in other areas to enable robot teaching.**

I. INTRODUCTION

Currently robot teaching is a necessary process in the industrial robot applications because industrial robots are typically lack of adaptability to the environmental changes and workpiece variations. Before a new application is deployed, offline programming and simulation packages such as RobotStudio [1] can be used to generate robot programs. Because of the unmodeled factors and parts' location errors, to transfer the virtually tested programs to the real application, a teaching process has to be performed to correct the errors in the work station. Even for robots in operation, when different batches of parts are launched, the pre-tuned robot tool locations may not be applicable because of the part variations. Thus the work station has to be stopped in order to correct the part location errors. Therefore the goal of this research is to investigate a fully automated system to enable industrial robots to situate themselves without human intervention such that the manufacturing efficiency can be increased.

Vision systems are widely used to correct the part location errors. However, the location output from the vision system may not be reliable because of the deviation of camera calibration. Hence directly using the vision system may not be able to correct the part location errors. Therefore, part location correction with uncalibrated vision systems is desirable.

Even though a vision system can be installed at each work station, the cost of n vision systems (n is the number of work stations) in a production line will be very high and the maintenance issues caused by many vision systems are significant. Hence installing a vision system on a simple mobile platform, instead of n vision systems, could reduce the cost and thus be a better solution to deal with the part location errors. A mobile robot equipped with cameras ("Camera-in-Mobile") is an "adult" robot that can teach a group of "child" robots in a production line. Since the "adult" robot has to move around, calibration between the camera and "child" robots is unrealistic. Therefore, the key difficulty of utilizing such a mobile "adult" robot lies in how to guarantee the teaching accuracy without calibration.

Without any prior knowledge, the mobile "adult" robot has to work in an unstructured environment. Motion control in such an environment is still a challenging problem since models of the working environment are not known as a prior. There are many methods proposed to solve this problem, such as 3D information retrieval including multi-view geometric techniques using single movable camera[2], [3] and Simultaneous Localization And Mapping(SLAM)[4]. However due to the limitation of vision systems, they cannot meet the requirements for highly precise and flexible robot teaching applications[5], [6]. Visual servoing with uncalibrated cameras[7], [8] has been proposed to solve the localization problem. Because of the strict requirements for the prior knowledge of robot dynamics and the environmental uncertainties such as lighting condition and part variations, visual servoing methods could generate big errors to command a robot and thus damage the system and/or parts.

In this paper, we utilize an "adult" robot with a 2D camera to realize the teaching process. By observing the uncertain workpiece features from the camera, formulating and solving the Partial Observable Markov Decision Process(POMDP)[9] model, the "adult" robot learns and teaches a "child" robot how to choose an action in the corresponding state. Experiments are performed to evaluate the proposed method. Because there is no calibration between the "adult" robot and the "child" robot, it is easy to deploy the developed system in robot teaching. Thus human workers can be released from the teaching work and the manufacturing efficiency can be improved. The proposed method can also be used for teaching processes using fixed cameras on a work station without calibration.

H. Cheng and L. Hao are with Department of Mechanical Engineering and Automation, North East University, Shenyang, Liaoning, China, {chenght,haolina}@me.neu.edu.cn

H. Chen is with the Ingram School of Engineering, Texas State University, San Marcos, USA, hc15@txstate.edu

W. Li is with Yantai Tongxing Industrial Co. Ltd., Yantai, Shandong, China, davidlee83@126.com

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

II. Proposed Solution

The proposed "adult" robot enabled teaching process is shown in Figure 1. This teaching process contains three subtasks: learning the primitive motion, offline robot learning and online robot teaching. The fourth task is about the assembly process.

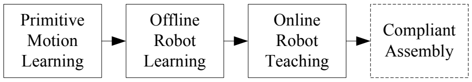


Fig. 1. Structure of the proposed robot enabled learning approach for robot teaching problem in the unstructured environment. The compliant assembly is a built-in function of the "child" robot.

Since there is no calibration between the "adult" and "child" robots, to ensure that the "child" robot moves along the right direction, the "adult" robot must learn its moving direction first. This subtask is performed using the following primitive motion learning algorithm.

A. Primitive Motion Learning

As mentioned above, the mobile "adult" robot has no prior knowledge about the coordinate frames of the "child" robot tool(C_t) and target(C_h). It has to learn how to guide the "child" robot first. The relationship between C_t and C_h drawn in the camera coordinate frame is shown in Figure 2. Since the goal is to drive the tool to the target position, it is natural to learn how to guide the "child" robot tool along the target coordinate frame C_h.

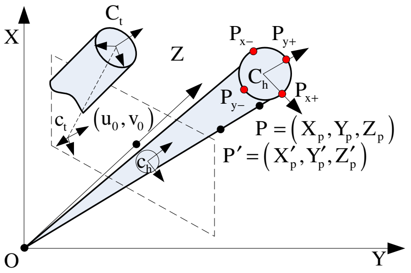


Fig. 2. Coordinate projection from the robot tool coordinate frame, target coordinate frame to the image 2D coordinate frame. c_h and c_t are the projected target and robot tool coordinate frames respectively. The view cone is a cone connecting the camera origin to the target. Objects in the cone will block the target. Four points $P_{x+}, P_{x-}, P_{y+}, P_{y-}$ are selected as the critical points for the block property.

The "adult" robot can command the "child" robot to offset along its tool coordinate frame C_t. To guide the "child" robot tool along C_h, the relationship between C_t and C_h should be determined. The difference between these two coordinated frames is determined by a rotation angle θ. Since it is difficult to obtain this angle in the original C_t and C_h, we calculate it in the projected coordinate frames c_t and c_h. By driving the "child" robot to move along its X-axis and recording the starting and ending positions (u_1, v_1) and (u_2, v_2) in the image coordinate frame one can get

$$[\Delta u \ \Delta v] = [u_2 - u_1 \ v_2 - v_1] \tag{1}$$

where $\Delta u, \Delta v$ are the projected tool position changes in the image coordinate frame. Therefore, the slope of the projected X-axis of C_t can be obtained by $k_x = \Delta u / \Delta v$. Furthermore, when the "adult" robot identifies the target, it can define the target coordinate frame C_h and its projection c_h. Therefore, the rotation angle θ can be calculated using the X-axis of the projected coordinate frames c_h and c_t.

$$\theta = \arctan k_{x_h} - \arctan k_x \tag{2}$$

where k_{x_h} is the slope of the X-axis of the projected target coordinate frame c_h. Thus, one can get the coordinate transform equation.

$$\begin{bmatrix} \Delta x_t & \Delta y_t \end{bmatrix}^T = M(\theta) \begin{bmatrix} \Delta x_h & \Delta y_h \end{bmatrix}^T \tag{3}$$

where $M(\theta)$ is the rotation matrix; $(\Delta x_t, \Delta y_t)$ are the command values sent to the "child" robot for realizing offset$(\Delta x_h, \Delta y_h)$ along the target coordinate frame.

B. Offline Robot Learning

To guide the robot tool to the desired position using 2D information, the position transformation from 3D to 2D should be analyzed. Based on the Pinhole camera model, the Z coordinate is missing during the projection process which means each single pixel (x, y) in the image coordinate frame stands for a 3D line in camera coordinate frame, i.e. $X = xZ/f, Y = yZ/f$, where f is the camera focal length and (X, Y, Z) are the coordinates in the camera coordinate frame. For the two points P and P' in Figure 2, although it is impossible to measure their actual coordinates, it is possible to determine whether $P'(X'_p, Y'_p, Z'_p)$ blocks $P(X_p, Y_p, Z_p)$.

1) POMDP: As aforementioned, to increase the system flexibility and reliability, the camera calibration is eliminated. Thus the only information we have is whether the critical points are blocked or not. Therefore the problem is how to estimate the underlying position errors and make correct decisions with limited observations, which meets the definition and requirements of POMDP [9].

The POMDP consists of a tuple $\{S, A, O, T, Z, R, \gamma\}$, where S is the state set; A is the action set; O is the observation set; T is the transition function; Z is the observation function; R is the reward function and $\gamma \in (0, 1)$ is the discount factor. Different from the standard Markov Decision Process(MDP), the state cannot be fully observed. Instead, an observation set O is introduced to describe the possible observations and observation function Z is introduced to describe the statistical relationship between O and the underlying state S.

2) Model Construction: For the robot teaching problem, the goal is to drive the "child" robot tool toward the target position. The state should be chosen according to the position error between the robot tool and target. Although the state space and robot manipulator dynamics are continuous, due to the limitation of the sensor and motion accuracy, the action and state are better represented using discrete variables.

It is natural to discretize the state space into a grid. Figure 3(a) shows a 2D Peg-in-hole example demonstrating the grid. The grid intersects with the view cone. Thus when the "child" robot moves the tool into this area, there always exist grids to block or unblock the target. The desired state lies exactly in front of the target as shown in Figure 3(a).

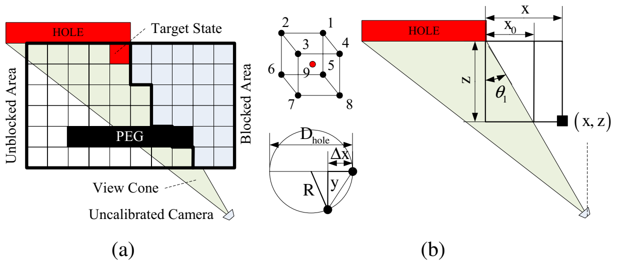


Fig. 3. a) A 2D grid and its corresponding view cone. The spatial space in front of the target hole is discretized into a grid. b) Geometrical model of the position errors and the block property. Where D_{hole} and R are the diameter and radius of the peg respectively. The condition for whether the critical point is blocked can be derived through this geometrical model.

The grid size can be chosen according to the actual requirements such as problem complexities, clearance, compliant motion accuracy, etc. From Figure 3(a) it is easy to verify that each grid may have different block properties.

To fully understand the relationship between the block property and the underlying position errors, the geometrical model has to be built. The critical points($P_{x+}, P_{x-}, P_{y+}, P_{y-}$) shown in Figure 2 are chosen to determine the block property.

Figure 3(b) is the top view of the view cone. Assuming that the tool is at the position (x,y,z) and $y = 0$, we can determine whether the critical point P_{x+} is blocked using

$$x > z \cdot \tan\theta_1 \tag{4}$$

where θ_1 is the view angle between edge point on the X-axis and the camera. If Equation (4) holds, the point is blocked; otherwise, it is unblocked. For the general case with $y \neq 0$, the offset along the Y-axis must be compensated. As shown in Figure 3(b), the block property determination criteria for P_{x+} becomes

$$x > z \cdot \tan\theta_1 + R - \sqrt{R^2 - y^2} \tag{5}$$

Similarly, the block determination criteria for P_{y+} is:

$$y > z \cdot \tan\theta_2 + R - \sqrt{R^2 - x^2} \tag{6}$$

where θ_2 is the view angle between edge point on the Y-axis and the camera. θ_1 and θ_2 can be obtained through visual pose estimation as proposed in [10].

Suppose the camera is at the position as shown in Figure 3(a), there are three basic observations: block_top means the the critical point P_{y+} is blocked, block_right means the the critical point P_{x+} is blocked; no_move means the tool is contact with the target(feedback from the "child" robot) and block_none means neither of these three are observed. Combining the basic observations together, we can get the following observations: block_none(o_0), no_move(o_1), block_top(o_2), block_top_no_move(o_3), block_right(o_4), block_right_no_move(o_5), block_right_top(o_6) and block_right_top_no_move(o_7).

Using Equations (5) and (6), the block property of a single point can be determined. However, no matter how small the grid is selected, it cannot be seen as a point. Therefore, the observation for each grid should be calculated carefully. We propose to use the following nine point method. Considering the cube shown in Figure 3(b), there are eight vertices and one center point. The observation of each point is assigned with different weight, hence the total observation for a grid follows

$$\begin{bmatrix} o_0(s) \\ o_1(s) \\ \vdots \\ o_7(s) \end{bmatrix} = \begin{bmatrix} \delta_{01} & \cdots & \delta_{08} & \delta_{09} \\ \cdots & \cdots & \cdots & \cdots \\ \delta_{61} & \cdots & \delta_{68} & \delta_{69} \\ \delta_{71} & \cdots & \delta_{78} & \delta_{79} \end{bmatrix} \begin{bmatrix} 1/16 \\ \vdots \\ 1/16 \\ 0.49 \end{bmatrix} + \begin{bmatrix} \varepsilon \\ 0 \\ \vdots \\ \varepsilon \\ 0 \end{bmatrix} \tag{7}$$

where $\delta_{ij} = \begin{cases} 1 & o_i = v_j \\ 0 & o_i \neq v_j \end{cases}$.

The weight of the center point is 0.49 while the weight of all other vertexes is $1/16$. $\varepsilon = 0.0025$ is a small bias value assigned to observation o_0, o_2, o_4, o_6 to overcome the possible observation errors. The purpose of ε will be illustrated in Section III B.

The action space is the set of movements along C_h named right($a_0 = x$+), left($a_1 = x$-), up($a_2 = y$+), down($a_3 = y$-) and forward($a_4 = z$+). Each action drives the "child" robot from one grid to another. The backward movements(moving along Z-axis in the negative direction) is removed because it is not necessary for the assembly problem. Therefore, totally there are five actions.

The goal is to drive the "child" robot to the position where the peg is approximately in front of the hole. If the goal is reached, the robot receives a big positive reward of +20; If there is a collision between the tool and the target, the robot receives a big negative reward -20; Additionally, it gets a reward of -1 on each step to encourage shorter paths.

3) Solving POMDP: Since the state is not fully observable, the agent does not know the exact state it is currently in. Instead, by executing actions and receiving the observations, it can maintain a probability distribution over the states S. For each state $s \in S$, the belief is denoted as $b(s)$. So totally, the belief state $\mathbf{b}$ is an $n \times 1$ vector where n is the number of states and $\sum_s b(s) = 1$. The belief state can be updated using the following equation,

$$b'(s'|b,a,o) = \frac{O(o|s')\sum_s b(s)T(s'|s,a)}{\sum_{s'} O(o|s')\sum_s b(s)T(s'|s,a)} \tag{8}$$

where $b'(s')$ is the new belief for state s'; a is the executed action and o is the received observation. Furthermore we can define the belief state transition function $\tau(\mathbf{b},a,o)$ as

$$\mathbf{b}' = \tau(\mathbf{b},a,o) = \begin{bmatrix} b'(s_1) & b'(s_2) & \cdots & b'(s_n) \end{bmatrix}^T \tag{9}$$

The new belief state $\mathbf{b}'$ only depends on the last belief state $\mathbf{b}$, which meets the Markov property. Therefore, the POMDP is converted into a continuous state MDP. Similar to defining the value function for each state in an MDP, we can define the value of the belief state $V(\mathbf{b})$ and the optimal value function $V^*(\mathbf{b})$ is the fixed point of the following equation:

$$V(\mathbf{b}) = \max_{a \in A}\left[r(\mathbf{b},a) + \gamma \sum_{o \in O} z(o|\mathbf{b},a) V(\tau(\mathbf{b},a,o))\right] \tag{10}$$

where $r(\mathbf{b},a) = \sum\limits_{s} b(s)\,T(s'|s,a)\,R(o|s')$ is the reward function; $z(o|\mathbf{b},a)$ is the observation function follows:

$$z(o|\mathbf{b}\,,a) = \frac{\sum\limits_{s} b(s)\,T(s'|s,a)\,Z(o|s')}{\sum\limits_{o}\sum\limits_{s} b(s)\,T(s'|s,a)\,Z(o|s')} \tag{11}$$

If the optimal value function is found, the optimal action for each belief state can be determined using

$$a^* = \arg\max_{a}\left[r(\mathbf{b},a) + \gamma \sum_{o} z(o|\mathbf{b},a)\,V^*(\tau(\mathbf{b},a,o))\right] \tag{12}$$

However, it is noticed that V is a function of an $n \times 1$ vector. $V(\mathbf{b})$ could be any form of functions and probably it is a nonlinear function. So it is very difficult to express the value function explicitly and accurately.

An alternative solution is to maintain a set of linear functions instead of maintaining a single nonlinear function. For finite-horizon POMDPs, the optimal value function is PieceWise-Linear and Convex(PWLC)[9]. Therefore, it can be approximated using a finite set of vectors Γ, i.e.,

$$V(\mathbf{b}) = \max_{\alpha\in\Gamma} V^{\alpha}(\mathbf{b}) \tag{13}$$

where

$$V^{\alpha}(\mathbf{b}) = \langle \alpha \cdot \mathbf{b} \rangle = \sum_{i} \alpha_i b_i$$

$$\alpha = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_n \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}^T$$

where α_i refers to $\alpha(s_i)$ and b_i refers to $b(s_i)$. The α vector represents factor of a linear function which is the locally linear approximation of the nonlinear value function. New α vector can be derived using the following backup operation(for detailed derivation, see[9]),

$$\begin{aligned} V'(\mathbf{b}) &= \max_{a\in A}\left(r_a \cdot \mathbf{b} + \gamma \sum_{o\in O} z(o|\mathbf{b},a)\,V(\tau(\mathbf{b},a,o))\right) \\ &= \max_{a\in A}\left(r_a \cdot \mathbf{b} + \gamma \sum_{o\in O, z(o|\mathbf{b},a)>0} \max_{\alpha\in\Gamma} \mathbf{b} \cdot \alpha^{a,o}\right) \end{aligned} \tag{14}$$

where $\alpha^{a,o} = \sum\limits_{s'\in S} \alpha(s')\,Z(a,s',o)\,T(s,a,s')$.

The problem is how to bound the size of the alpha vector set. In the robot teaching problem, thousands of states may exist if the accuracy requirement is high. Therefore, solving the complex POMDP problem and searching for an optimal policy are difficult tasks. There is a contradiction between accurate approximation and compact representation of the value function. The solution of this problem is to select a set of belief points denoted as B, and maintain only vectors that are optimal for at least one belief point in the set, known as the Point-Based Value Iteration(PBVI) algorithm proposed in 2003[11].

The PBVI algorithm explores the belief space, focusing on the reachable belief states, while maintaining a value function by applying the point-based backup operator, which can greatly increase the computational efficiency. Since then, a lot of improvements has been made and current solvers like PBVI[11], Heuristic Search Value Iteration(HSVI)[12] and Successive Approximation of the Reachable Space under Optimal Policies(SARSOP)[13] are capable of handling complex domains with many thousands of states.

C. Online Robot Teaching

Once POMDP is solved and the optimal policy is found, the "adult" robot is able to guide the "child" robot tool toward the target position. At each step, the "adult" robot computes the best action, sends it to the "child" robot to execute, observes the tool location and updates its belief state. The one step look ahead action selection algorithm follows Equation (12). By iteratively performing actions and observations, the belief state is updated and the "adult" robot gradually learns the real location of the "child" robot. And finally, the "child" robot is guided to the target position.

III. Experiments and Results

To verify the effectiveness of the proposed algorithms, we implemented the robot enabled teaching process using the platform shown in Figure 4.

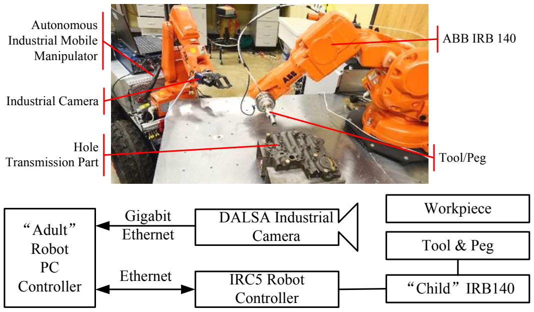


Fig. 4. The platform for automatic robot teaching with camera-in-mobile configuration. The "child" robot is performing precise assembly task.

The platform is consisted of two robots. The ABB IRB140 is fixed on the work cell while the ABB IRB120 is mounted on a mobile platform. The IRB140 is the "child" robot and driven directly by ABB IRC5 robot controller. The assembly task is to insert a peg into the hole on a transmission part. The tolerance between the peg and hole is 40μm. Therefore, it is difficult to perform the assembly task using pure position based control. The diagram in Figure 4 shows the configuration of the experimental system. The DALSA industrial camera (Frame Rate: 100Hz, Resolution:1024×768), the camera, the "child" robot and "adult" robot controller are connected via Ethernet. An image captured by the industrial camera is shown in Figure 5.

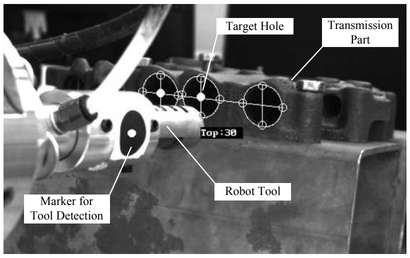


Fig. 5. Captured image from the camera. It is a 2D image with no depth information. The goal is to insert the peg into the middle hole.

In these experiments, we considered the most common scenario: the "adult" robot comes, stands still and performs a teaching task. During the experiments, the camera and the mobile robot are fixed and there is no calibration between the camera, the "adult" robot and the "child" robot.

A. *Experimental Procedure*

The whole assembly process contains four subtasks: primitive motion learning, offline robot learning, online robot teaching and compliant assembly. These subtasks are integrated to accomplish the assembly task. During the primitive motion learning stage, the "adult" robot firstly sends commands to the "child" robot with offsets $\Delta x = 5$mm, records the starting and ending position of the "child" robot tool and calculates the primitive motion transformation matrix.

1) Offline Robot Learning: To obtain the policy for controlling the "child" robot, the POMDP model has to be generated first. Parameters of the system configuration are listed in TABLE I. We consider two groups of parameters with different grid sizes.

TABLE I

POMDP Model Parameters

g_x	g_y	g_z	N_X	N_Y	N_Z	D_{hole}	θ_1/rad	θ_2/rad
5mm	5mm	5mm	8	8	8	22mm	0.67	-0.49
2mm	2mm	2mm	16	16	16	22mm	0.67	-0.49

The discount factor γ is chosen as 0.95. The POMDP model is solved using the Approximate POMDP PLanning toolkit(APPL) solver (http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/) which implements the SARSOP algorithm.

Here the initial state is assumed to the states which have the observation of o_2 and are 20mm away(along Z-axis) from the target position. For the 5mm grid POMDP model, it takes about 10.2s to find the optimal policy while for the 2mm grid POMDP model, the solving process lasts 3003.1s.

2) Online Robot Teaching: To Verify the effectiveness of the proposed robot teaching algorithm and the accuracy of the obtained POMDP policy, we utilized the parameters shown in TABLE II to generate the random initial positions. The case 1 parameters are for 5mm grid configuration experiments and case 2 parameters are for 2mm grid configuration experiments.These initial positions follow the normal distribution and satisfy the initial observation of the aforementioned POMDP(Block only the top critical point). Where $\mu_.$, $\sigma_.$ and $B_.$ are the mean, the standard deviation and the bounds of each variable. These points are also plotted in Figure 6(b) and 8(b). After performing the above offline optimizing process, the "adult" robot is able to teach the "child" robot. The "adult" robot selects the action greedily according to the one step look ahead algorithm and send it to the "child" robot until the "child" robot reaches the desired position.

TABLE II

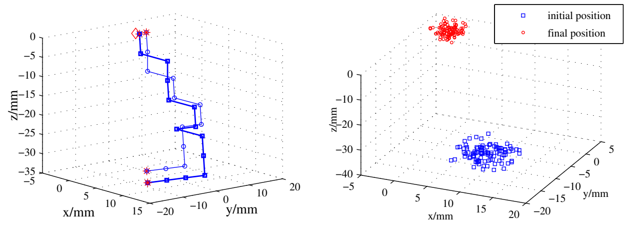
Parameters for Generating the Initial Position

case	μ_x	σ_x	B_x	μ_y	σ_y	B_y	μ_z	σ_z	B_z
1	10	3	[6,16]	-14	3	[-21,-7]	-29	1	>-40
2	10	3	[6,16]	-7	3	[-13,-2]	-29	1	>-32

3) Compliant Assembly: The "child" robot is preprogrammed with compliant assembly algorithm proposed by Chen [14]. The assembly process is divided into three stages: searching, insertion and settle-down. The controller searches the hole position along a spiral curve with a 2mm search radius and inserts the tool into the hole using a force control method. Therefore, the goal of the robot teaching is to guide the robot tool close enough to the target hole within the accuracy of 2mm. A more accurate final position will lead to a more efficient assembly process.

B. *Experimental Results*

1) Experiments with 5mm grid size: We first tested the policy generated using the 5mm grid size. The tool trajectories are plotted in Figure 6(a). The tool started from random initial positions and was guided toward the target using the learned primitive motions. In each step, the tool moved 5mm along the selected moving direction. It took about 14 steps for the "child" robot to arrive at the target location.



(a) (b)

Fig. 6. Experiments of the 5mm grid size. a) Two robot tool motion trajectories. The diamond is the target position. b) Distribution of the initial positions and final positions.

Using randomly generated initial positions, we repeated this process for 100 times. The initial and final positions are plotted in Figure 6(b). The "child" robot was successfully guided to the target. However, since the grid size is 5mm, the final tool position has a 5mm variation along each axis. The statistic results are listed in TABLE III where the case 1 is for 5mm grid configuration and case 2 is for 2mm grid configuration.

TABLE III

Statistical Results of the Final Robot Tool Position

case	μ_x	σ_x	μ_y	σ_y	μ_z	σ_z
1	-2.10	1.29	3.02	1.22	-3.42	1.41
2	-0.97	0.51	1.03	0.77	-1.01	0.52

During the teaching process, the "adult" robot updated its belief state based on the action and observation. Figure 7 shows the maximus belief at each step. We can find that the belief is increasing when more and more actions-observations are made.

2) Experiments with 2mm grid size: In order to improve the teaching accuracy, we decreased the grid size from 5mm to 2mm. And thus the system dimension has to be doubled to cover most of the initial positions. Figure 8(a) shows some of the "child" robot tool trajectories. The "child" robot moved about 25 steps to reach the target. We repeated this process for 100 times and the initial and final positions are plotted in Figure 8(b).

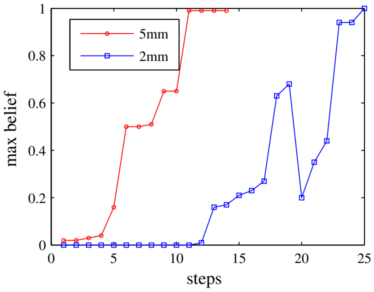
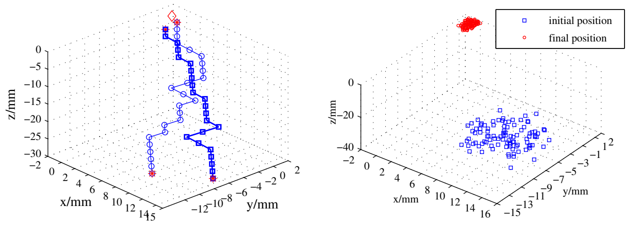


Fig. 7. Maximum belief during the teaching process. Ideally the maximum belief is nondecreasing. However, due to noises and errors, there is a sudden drop in one of the maximum belief curve.



(a) (b)

Fig. 8. Experiments of the 2mm grid size. a) Two robot tool motion trajectories. The diamond is the target position. b) Distribution of the initial positions and final positions.

The statistical results of the final position for 2mm grid size is listed in TABLE III.

3) Discussion: According to the above results, small grid size can achieve better accuracy. However, when the grid size becomes smaller, it imposes strict computational and accuracy requirements for the POMDP observation and state transition model. If the real environment does not match the model based on which the policy is generated, the “adult” robot may not be able to find any possible state, i.e. $\sum_{s\in S} b(s) = 0$, which contradicts with the definition of the belief state and makes the algorithm fail. To deal with the problem, we introduced ε in the observation model (7). Once a “wrong” observation is received, the maximum belief will drop but the belief state still satisfies the definition. However, utilizing a robust observation model and decreasing the grid size to meet the accuracy requirements greatly increase the computational complexity.

This work is an initial attempt to perform robot teaching with a single uncalibrated camera. The experimental results verify the effectiveness of the POMDP modeling and SARSOP solving algorithm and also demonstrate that the obtained optimal policy can deal with uncertainty and gather useful information for decision making with limited observations.

IV. Conclusion

This paper discusses the problem of robot teaching in unstructured industrial robot applications using an uncalibrated vision system. We propose to use a mobile “adult” robot holding a camera to move around and teach a group of robots working in a production line. To eliminate the calibration process and reduce the restrictions between the sensors and robots, a POMDP is formulated to estimate the underlying position errors and make decisions at each step. The primitive motion learning, POMDP based offline learning, online robot teaching and compliant assembly subtasks are integrated to accomplish a successful robot teaching and assembly process. Experimental results show the effectiveness of the proposed method in robot teaching using vision systems with uncertainties. Because the calibration process is eliminated, the flexibility of the proposed method is greatly increased. Hence it can be easily applied in industrial applications where robot teaching is needed. The proposed method can also be used for robot teaching using fixed cameras on a work station without calibration.

Acknowledgment

The research is partially sponsored by the Research Enhance Program(REP). Grant No.9000000936, Texas State University, San Marcos. And it is also partially sponsored by the Startup Research Fund. Grant No.02090021233043, Northeastern University, China.

References

[1] C. Connolly, “Technology and applications of abb robotstudio,” *Industrial Robot: An International Journal*, vol. 36, no. 6, pp. 540–545, 2009.

[2] A. Kundu, K. Krishna, and J. Sivaswamy, “Moving object detection by multi-view geometric techniques from a single camera mounted robot,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 4306 –4312.

[3] G. Hu, W. MacKunis, N. Gans, W. Dixon, J. Chen, A. Behal, and D. Dawson, “Homography-based visual servo control with imperfect camera calibration,” *Automatic Control, IEEE Transactions on*, vol. 54, no. 6, pp. 1318 –1324, june 2009.

[4] A. Davison, I. Reid, N. Molton, and O. Stasse, “Monoslam: Real-time single camera slam,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1052 –1067, june 2007.

[5] T. Murao, H. Kawai, and M. Fujita, “Passivity-based control on dynamic visual feedback systems with movable camera configuration,” *Electronics and Communications*, vol. 92, no. 6, pp. 286–294, 2009.

[6] V. Lippiello, B. Siciliano, and L. Villani, “Eye-in-hand/eye-to-hand multi-camera visual servoing,” in *IEEE Conference on Decision and Control, European Control Conference*, dec. 2005, pp. 5354 – 5359.

[7] M. Marshall, M. Matthews, A.-P. Hu, G. McMurray, and H. Lipkin, “Uncalibrated visual servoing for intuitive human guidance of robots,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, may 2012, pp. 4463 –4468.

[8] P. Goncalves, L. Mendonca, J. Sousa, and J. Pinto, “Uncalibrated eye-to-hand visual servoing using inverse fuzzy models,” *Fuzzy Systems, IEEE Transactions on*, vol. 16, no. 2, pp. 341 –353, april 2008.

[9] G. Shani, J. Pineau, and R. Kaplow, “A survey of point-based pomdp solvers,” *Vital And Health Statistics. Series 20 Data From The National Vitalstatistics System Vital Health Stat 20 Data Natl Vital Sta*, 2012.

[10] G. Wang, Q. Wu, and Z. Ji, “Pose estimation from circle or parallel lines in a single image,” in *Computer Vision C ACCV 2007*, ser. Lecture Notes in Computer Science, Y. Yagi, S. Kang, I. Kweon, and H. Zha, Eds. Springer Berlin / Heidelberg, 2007, vol. 4844, pp. 363–372.

[11] J. Pineau, G. Gordon, and S. Thrun, “Point-based value iteration: An anytime algorithm for pomdps,” 2003.

[12] T. Smith and R. Simmons, “Heuristic search value iteration for pomdps,” in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, ser. UAI ’04. Arlington, Virginia, United States: AUAI Press, 2004, pp. 520–527.

[13] H. Kurniawati, D. Hsu, and W. S. Lee, “Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces,” in *In Proc. Robotics: Science and Systems*, 2008.

[14] H. Chen, J. Wang, G. Zhang, and T. Fuhlbrigge, “Robotic soft servo for industrial high precision assembly,” *Assembly*, pp. 24–29, 2008.