A Robot Self-learning Grasping Control Method Based on Gaussian Process and Bayesian Algorithm

Yong Tao, Hui Liu, Xianling Deng, Youdong Chen, Hegen Xiong, Zengliang Fang, Xianwu Xie and Xi Xu

Abstract A robot self-learning grasping control method combining Gaussian process and Bayesian algorithm was proposed. The grasping gesture and parameters of the robot end-effector were adjusted according to the position and pose changes of target location to realize accurate grasping of the target. Firstly, a robot self-adaptive grasping method based on Gaussian process was proposed for grasping training in order to realize modeling and matching of position and pose information of target object and robot joint variables. The trained Gaussian process model is combined with Bayesian algorithm. The model was taken as priori knowledge and the semi-supervised self-learning was implemented in new grasping region so that posterior Gaussian process model was generated. This method omits the complex visual calibration process and inverse kinematics solves only with a small group of samples. Besides, when the environment of grasping changes, the previous learning experience can be used to perform self-learning, and adapt to the grasping task in the new environment, which reduces the workload of operators. The effectiveness of the robot self-learning grasping control method based on Gaussian process and Bayesian algorithm was verified through simulation and grasping experiment of UR3.

Keywords Gaussian process · Bayesian algorithm · Robot grasping
Semi-supervised self-learning

Y. Tao · H. Liu (✉) · Y. Chen · Z. Fang
Beihang University, Beijing 100191, China
e-mail: huiliu@buaa.edu.cn

X. Deng
Chongqing University of Science and Technology, Chongqing 401331, China

H. Xiong · X. Xie · X. Xu
Wuhan University of Science and Technology, Wuhan 430081, China

S. Chen et al. (eds.), *Transactions on Intelligent Welding Manufacturing*,
Transactions on Intelligent Welding Manufacturing,
https://doi.org/10.1007/978-981-10-8330-3_5

1 Introduction

With maturation of robot technology in daily life, robots have entered people's life and production more and more frequently. In industrial production, it's a quite common application to operate manipulators to grasp target object such as transporting goods, systemizing articles and assembling parts which contain the most basic pick-and-place tasks. The grasp algorithm is also a major research hotspot in present robot orientation [1–6]. In operation of robot grasping articles, the position and pose of the grasped target object are usually not fixed. The self-adaptive grasping should be completed by adjusting grasping posture of robot end-effector according to position and pose of the target object.

According to different requirements for self-adaptive grasping of target object, some studies have added tactile sensors at end-effectors. Based on implemented encoding and decoding analysis according to contact information fed back by sensors, the grasping effects have been evaluated and motion parameters of the robot have been adjusted to complete favorable grasping [7–10]. In the Ref. [8], a probabilistic learning method for evaluating the stability of the grasp has been proposed. According to the tactile sensor feedback information, the performance of grasp is evaluated, so that objects can be regrasped before attempting to further manipulate them. In Refs. [9, 10], a hierarchical mechanism of two-step grasping is established. In the first step, based on tactile feedback, a grasp stability predictor is trained by supervised learning to predict performance of the grasp. In the second step, the parameters of grasp action are finely tuned according to the tactile sensor feedback. The methods above can improve adaptability of end-effector. However, it can only realize self-adaptive grasping nearby working region of the end-effector, and the posture adjustment by tactile perception is adverse to real-time grasping operation.

More studies have used visual sensors to obtain objects' position and pose to adjust the motion of robot and completed self-adaptive grasping of target object. The research methods of using visual sensor are divided into analytical methods and learning-based methods. Analytical methods use the image and depth information of the observed objects to model reshaping and point cloud segmentation, and then match a 3D model for each segmented object and analysis to obtain their pose information. Combined with the physical characteristics of each operation object and a grasp quality metric [11, 12], simulation would be performed in the physical simulator. Ultimately, suitable action would be chosen to perform the adaptive grasp of the objects [13–15]. However, the methods above have the following defects: firstly, complete model parameters are needed to pre-model the grasping objects which increase the workload. Secondly, it's difficult for the present depth and visual sensors to obtain complete model information and errors will easily occur during model matching process. The error exists between simulation result in simulator and actual operation result so that reliability of grasping algorithm will be greatly degraded. Thirdly, these methods need visual calibration for depth visual system of the robot. The traditional robot visual system calibration method has high requirements for professional knowledge and operating skills of working personnel [16–18], which consumes more time.

In the study of learning-based methods, the deep-learning technology is used. By constructing the deep-learning network, the mapping from images to action parameters of robot is established. The entire training process does not require any human intervention or model parameters and pre-modeling. Meanwhile, the calibration of camera is omitted, and it has stronger generalization ability and robustness. In Refs. [19, 20], manipulators collected data through self-supervised learning and trained a large Convolutional Neural Network (CNN) to generate the optimal grasping according to the visual information. In Ref. [21], CNN is trained with LSTM. The network uses unlabeled data to perform an end-to-end learning which established a predictor for estimating grasping effect. However, the application of deep-learning and neural network technology in learning-based methods requires a large number of training samples, and the training process is very time-consuming. Meanwhile, a long time to collect samples and training will increase wear and tear of hardware. How to get good learning effects through less training samples is a question worthy of study.

The visual servo system combines robot control and vision together without the need for pre-modeling and visual calibration [22, 23]. Indrazno [24] uses visual servo to achieve adaptive position control of a 7-DOF robot. Thomas [25] uses the visual servo to control the aircraft to complete the grasp action and the docking action. In Ref. [26], visual servo controller combined with reinforcement learning is applied to a mobile robot with a manipulator, which shows a good performance in robust grasping tasks. The use of visual servo can achieve the robot to adapt to the target object pose. However, the adjustment process of robot movement needs to analyze and deal with the image continuously, which increases the computational burden of the system and reduces the efficiency of the system.

In order to avoid solving the inverse solution and simplify the motion planning process, some studies teach robot to complete the grasping task by demonstration. In Ref. [27], the robot is taught to grasp different kind of objects by demonstration. The grasp type and the thumb position of each demonstration are recorded as the label of the corresponding grasp task. In Ref. [28], iterative learning is applied to the robot adaptive grasping control. In the learning process, adding manual adjustment of the robot's operating parameters to the demonstration, the robustness of the grasping control is increased. Ref. [29] proposed an object centred probabilistic volumetric model used to combine the multimodal data in the demonstration. The feature extracted by this method is proved to be useful for segmentation of the action phases and trajectory classification. In the process of demonstration, in order to generalize the executable tasks and improve the adaptability of the grasp task, it is particularly important to choose a mapping model from the observations to the motion parameters. As a Bayesian learning method, Gaussian process method (GPM) [30] can give robots the ability to learn the mapping function from the samples. Only small group of samples are needed to complete the training of the GPM and construct the nonlinear relation between the relevant variables. To solve the problem of physical Human-Robot interaction, Ghadirzadeh [31] uses GPM to establish the map between state-action pair and the variation of observations. In Ref. [32], in order to build a visual forward model, GPM is used to establish the mapping relation between motor commands and image observations. The researches above show that GPM is a powerful tool for non-parametric and non-linear regression.

In general, the requirements of environment for visual servo system are strict. This means that if the relative position of the robot and camera slightly changes, or if a grasping task is performed in a new and uncalibrated area, the accuracy of the grasp will be poor. Having this problem, in the application process whenever the environment changes, operators have to repeat the cumbersome calibration of visual system, which is time-consuming and increase the workload of operators. As for the GP method, under the circumstance of small sample size, it's not enough to rely only on GP probabilistic prediction model as it can obtain favorable generalization ability only nearby training samples. When the grasping environment changes or in a new region, the error also tends to increase. In order to adapt the GP model to the changing environment and to expand the scope of adaptive grasping, Bayesian method is adopted to combine with GPM as a self-learning algorithm.

A robot self-adaptive grasping algorithm based on Gaussian process was proposed. The position and pose information of target object obtained through visual grasping and corresponding robot joint variables were associated. It's only necessary to let the robot learn from demonstrated samples under small sample size which omitted calibration of robot visual system and inverse kinematics solving. Then the robot self-learning grasping control method based on Gaussian process and Bayesian algorithm was presented to do semi-supervised self-learning grasping in a new grasping region to generate the posteriori GP model, which improved adaptability of robot grasping. Grasping experiment of UR robot proved that robot self-learning grasping control method based on Gaussian process and Bayesian algorithm was of favorable effect.

2 Self-adaptive Grasping Based on Gaussian Process

2.1 Task Modeling

When executing a grasping task, the robot needs to match its own joint angle parameters and adjusts its motion according to position information of the target object. Self-adaptive grasping model of the robot to the target object is generalized as below:

$$f : \boldsymbol{o} \rightarrow \boldsymbol{a} \tag{1}$$

where $\boldsymbol{o}$ represents observations of target object, $\boldsymbol{a}$ is the corresponding joint coordinate, and f is mapping function from observation variable of the target object to joint coordinate.

It is assumed that is $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ the sample set obtained through demonstration, where $\boldsymbol{x}_i = [\boldsymbol{a}_i, \boldsymbol{o}_i]^T$ represents sample vector consisting of joint variables and observations. Robot self-adaptive grasping process lies in learning from sample set $\boldsymbol{X}$ to obtain mapping function f so that the robot can obtain the corresponding joint coordinate $\boldsymbol{a}$ according to the observation $\boldsymbol{o}$.

2.2 System Description

As shown in Fig. 1, the right part is experiment platform, left part is UR manipulator of six degrees of freedom, and an industrial camera is fixed right above the experimental platform. The camera optic axis is vertical with experimental platform, and the object is located within visible scope of the camera on the experimental platform. Two coordinate systems in the figure are respectively $\{B\}$ manipulator basic coordinate system and $\{T\}$ tool coordinate system.

Within a certain training region, the target object is arbitrarily placed, the industrial camera acquires pixel coordinates of the target location. The demonstration is then given, the most appropriate robot joint angle is selected so that the robot end-effector can execute grasping task accurately. The robot joint and observation of pixel coordinates of the target location are established as the sample sets. With information from training sets, Gaussian process model is used to establish mapping relation between the two sets. In the experiments, the robot completes the grasping task through robot joint angle correspondingly predicted through Gaussian process according to position information of target object obtained by the camera.

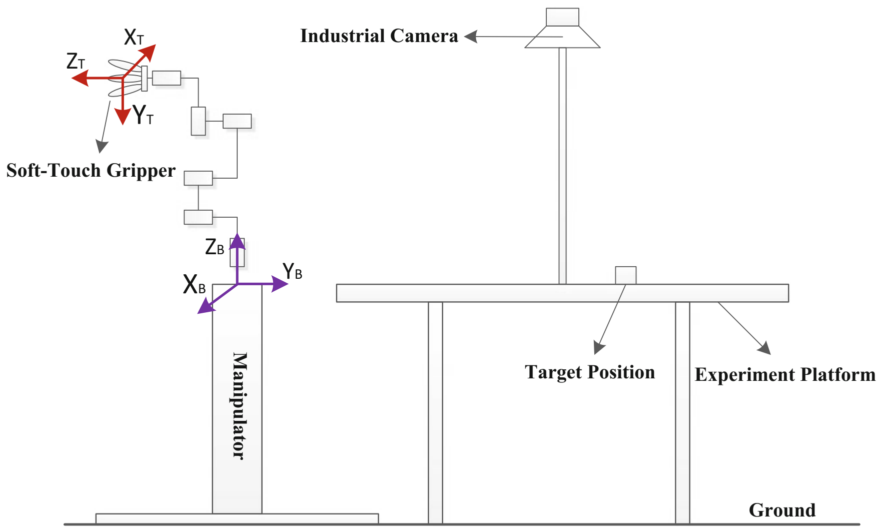

Fig. 1 The platform of manipulator grasping

2.3 Gaussian Process Model

GPM is a nuclear learning machine with probabilistic meaning, which can give a probabilistic interpretation of the predicted output. GPM is based on the assumption that the observations and predictions are subject to a joint normal distribution, then the posterior distribution of the predictions would be obtained by solving the covariance matrix of the observations and the input of the training set. GPM has been applied to the regression and classification problems successfully [10, 11]. The robot self-adaptive grasping method based on Gaussian process is as shown in Fig. 2. The method has omitted calibration of visual system and inverse kinematics solving. The robot needs to learn from samples to obtain parameters of Gaussian process model.

Before data are obtained, it's assumed that joint variable and observational variable of the target object comply with Gaussian distribution with mean value $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{K}$:

$$\boldsymbol{h} \sim (\boldsymbol{\mu}, \boldsymbol{K}) \tag{2}$$

In the equation, $\boldsymbol{h} = [\boldsymbol{a}, \boldsymbol{o}]^{\boldsymbol{T}}$ represents vector consisting of observation and joint variables. Sample set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{\boldsymbol{n}}\}$ obtained through demonstration includes measurement noise, and then:

$$\boldsymbol{x}_{\boldsymbol{i}} = \boldsymbol{h} + \varepsilon \tag{3}$$

In the equation, ε represents Gaussian noise with mean value 0 and variance σ_n^2.

Posteriori distribution of multidimensional variables obtained through sample set $\boldsymbol{X}$ is also Gaussian distribution:

$$p(\boldsymbol{h}|\boldsymbol{X}, \boldsymbol{\theta}) = N\left(\boldsymbol{\mu}, \boldsymbol{K} + \sigma_n^2 \boldsymbol{I}\right) \tag{4}$$

where $\boldsymbol{\theta} = \left\{\boldsymbol{\mu}, \boldsymbol{K}, \sigma_n^2\right\}$. For sample set $\boldsymbol{X}$, its marginal likelihood function is:

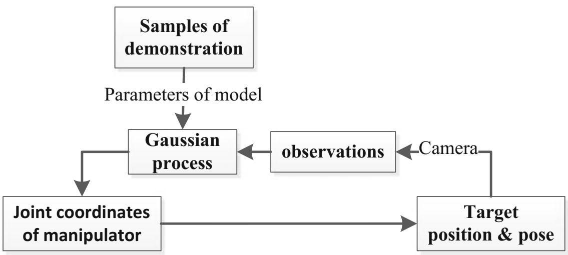

Fig. 2 Adaptive grasping based on Gaussian process

$$p = (\boldsymbol{X}|\boldsymbol{H}, \boldsymbol{\theta}) = \prod_{i=1}^{n} p(\boldsymbol{x_i}|\boldsymbol{h_i}, \boldsymbol{\theta}) = \prod_{i=1}^{n} p(\boldsymbol{x_i}|\boldsymbol{h_i})p(\boldsymbol{h_i}|\boldsymbol{\theta}) = \boldsymbol{P} \tag{5}$$

where $\boldsymbol{P} = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\left|\boldsymbol{K} + \sigma_n^2\boldsymbol{I}\right|}} \exp\left(-\frac{\tau_i}{2}\right)$, and $\tau_i = (\boldsymbol{x_i} - \boldsymbol{\mu})^T \left(\boldsymbol{K} + \sigma_n^2\boldsymbol{I}\right)^{-1}(\boldsymbol{x_i} - \boldsymbol{\mu})$

Partial derivatives of mean value vector and covariance matrix of the model are respectively solved through Eq. (5), derivatives are set as 0. The maximum likelihood estimation values of mean value vector and covariance matrix of Gaussian process can be obtained respectively as below:

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^{n} \boldsymbol{x_i}}{n} \tag{6}$$

$$\left(\boldsymbol{K} + \sigma_n^2\boldsymbol{I}\right) = Cov\left([\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x_n}]^T\right) \tag{7}$$

2.4 Prediction of Joint Variables

The following is obtained by blocking vector and matrix of Gaussian process:

$$\begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{o} \end{bmatrix} \sim N\left[\begin{bmatrix} \boldsymbol{\mu_a} \\ \boldsymbol{\mu_o} \end{bmatrix}, \begin{bmatrix} \boldsymbol{K_{aa}} + \sigma_n^2\boldsymbol{I} & \boldsymbol{K_{ao}} \\ \boldsymbol{K_{oa}} & \boldsymbol{K_{oo}} + \sigma_n^2\boldsymbol{I} \end{bmatrix}\right] \tag{8}$$

The robot obtains observation information $\boldsymbol{o}^*$ of the target object from the camera, and then corresponding conditional probability distribution of joint angle $\boldsymbol{a}^*$ is:

$$p(\boldsymbol{a}^*|\boldsymbol{o}^*) = N\left(\boldsymbol{\mu}_{\boldsymbol{a}}^*, \boldsymbol{K}_{\boldsymbol{aa}}^*\right) \tag{9}$$

where $\boldsymbol{\mu}_{\boldsymbol{a}}^* = \boldsymbol{\mu_a} + \boldsymbol{K_{ao}}\left(\boldsymbol{K_{oo}} + \sigma_n^2\boldsymbol{I}\right)^{-1}(\boldsymbol{o}^* - \boldsymbol{\mu_o})$, $\boldsymbol{K}_{\boldsymbol{aa}}^* = \boldsymbol{K_{aa}} - \boldsymbol{K_{ao}}\left(\boldsymbol{K_{oo}} + \sigma_n^2\boldsymbol{I}\right)^{-1}\boldsymbol{K_{oa}}$.

$\boldsymbol{\mu}_{\boldsymbol{a}}^*$ is mean value of joint angle matching new target location and it's corresponding to maximum probability of Gaussian distribution. $\boldsymbol{K}_{\boldsymbol{aa}}^*$ is covariance matrix of Gaussian distribution and it represents uncertainty of prediction result. Grasping can be completed by the robot at maximum probability by driving robot joint to reach $\boldsymbol{\mu}_{\boldsymbol{a}}^*$.

As it's not necessary to do visual system calibration and inverse kinematics solving, Gaussian process directly associates robot joint variables and observational variables of the target object. According to new observations, the robot joint angle corresponding to position of the target object is predicted.

3 Semi-supervised Self-learning Grasping Based on Bayesian Algorithm

GP method performs well in the training region with only small dataset. The method is of decent generalization ability. However, some errors may exist in this method when it used for grasping beyond training region. In order to expand effective grasping scope of manipulator, the Bayesian algorithm is adopted. The GP model trained before is used as priori model which is then added into semi-supervised learning process. New training samples are collected after grasping training in new adjacent region through robot self-learning grasping so as to update probability distribution of the whole Gaussian process. The posteriori probability model is obtained.

As shown in Fig. 3, target position is randomly selected in new training region as input of GP model. Based on Gaussian self-adaptive strategy discussed in Sect. 2, the GP model is used as priori model to generate robot joint angle parameters. In supervised learning process, the solution of the forward kinematic of the manipulator is solved according to the joint angle. Then a posture evaluation mechanism is used for feedback and fine adjustment of joint angle. After a relatively reasonable terminal grasping posture is obtained, the manipulator will try grasping, and the actual grasping position of end-effector is acquired. On the one hand, joint angle parameters and corresponding actual grasping position are taken as new samples in training region to update the training set. On the other hand, according to actual grasping position and target location, grasping evaluation mechanism is used for evaluation.

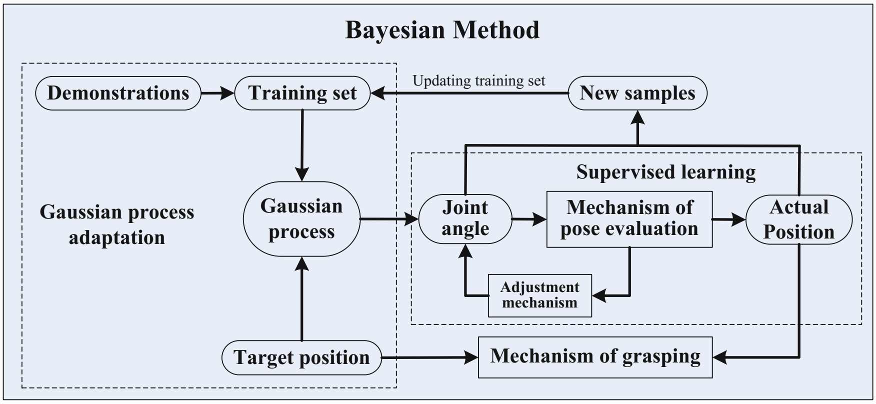

Fig. 3 Semi-supervised self-learning control method based on Bayesian method

3.1 Semi-supervised Learning

During demonstrated sampling process, collected sample posture is set as fixed posture. The gripper executes grasping from up to down in the direction vertical to plane of the experimental platform as shown in Fig. 4. Z_T axis of tool coordinate system is parallel to and in reverse direction to Z_B axis. Correspondingly, $X_T O_T Y_T$ plane is parallel to $X_B O_B Y_B$ plane. H is the height of grasping location of the target object under $\{B\}$ robot reference coordinate system. During semi-supervised learning process, robot joint angle is generalized according to target location through Gaussian process. In order to guarantee effectiveness of successful grasping, final grasping posture of end-effector is made to approach posture in demonstration samples as much as possible. Meanwhile, the height of grasping location should approach height of end-effector in demonstration samples as much as possible. Adjustment is completed through two iterative loops, and concrete operation is as below:

According to pixel coordinate of objects in the new adjacent training region, the self-adaptive grasping method based on Gaussian process in Sect. 2 is used to predict joint angle vector $\boldsymbol{\mu}_a^*$ with maximum successful grasping rate. The D-H modeling of the manipulator is implemented and joint angle vector $\boldsymbol{\mu}_a^*$ is substituted into forward solution formula of robot kinematics to obtain coordinate transformation matrix.

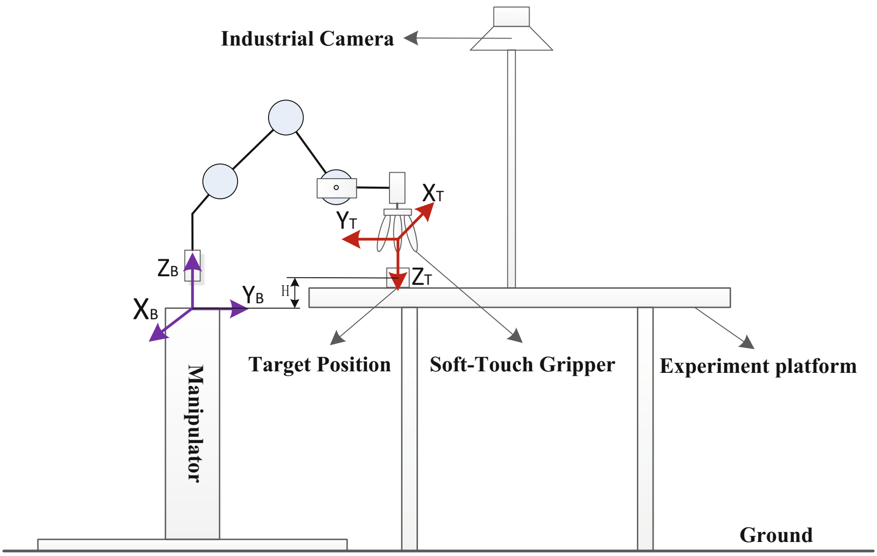

Fig. 4 The pose of end-effector

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\boldsymbol{\theta}_z$ is used to express relationships between Z_T axis in tool coordinate system and axes of reference coordinate system. P represents displacement of origin of tool coordinate system in direction of Z_B axis under reference coordinate system.

$$\boldsymbol{\theta}_z = \begin{bmatrix} \frac{\pi}{2} - \arccos(r_{13}) \\ \frac{\pi}{2} - \arccos(r_{32}) \\ \pi - \arccos(r_{33}) \end{bmatrix}; \quad P = p_z - H$$

where $\theta_{zi} \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \ i = 0, 1; \quad \theta_{zi} \in [0, \pi], \ i = 2.$

According to obtained $\boldsymbol{\theta}_z$, loop iteration A is implemented and robot joint angle is adjusted.

$$\boldsymbol{\mu}_a = \boldsymbol{\mu}_a + \boldsymbol{w}\boldsymbol{\theta}_z^T$$

where $\boldsymbol{\mu}_a$ is the robot joint angle, and $\boldsymbol{w}$ is correction coefficient matrix.

Termination conditions of iterative loop A are:

$$\begin{cases} \text{Execute loop A and enter loop B} & |\theta_{zi}| < \delta, i = 0, 1, 2 \\ \text{Execute loop A} & \text{else} \end{cases}$$

After termination of iterative loop A, joint angle parameters with reasonable grasping posture are obtained to do iterative loop B so as to adjust robot joint angle.

$$\boldsymbol{\mu}_a = \boldsymbol{\mu}_a + \boldsymbol{w}'P$$

where $\boldsymbol{w}'$ is correction coefficient vector of each joint angle.

Termination conditions of iterative loop B are:

$$\begin{cases} \text{Goto loop A} & |P| < h \text{ and } |\theta_{zi}| \geq \delta, i = 0, 1, 2 \\ \text{Execute loop B,} & |P| \geq h \\ \text{End loop B,} & \text{else} \end{cases}$$

After termination of iterative loop B, joint angle parameters with both reasonable grasping posture and grasping height are obtained. Execute grasping operation and observe the grasping position of the gripper which is taken as new training sample together with joint angle parameters. The training sets of GP are updated.

3.2 Bayesian Algorithm

The Bayesian algorithm has been proposed to do modeling of grasping in new training region. Main idea of Bayesian algorithm can be expressed by the following equation:

$$p(\theta|X) = \frac{p(X|\theta) \cdot p(\theta)}{p(X)}$$

where $p(\theta)$ is probability distribution of priori model, X is training sample, $p(\theta|X)$ is probability distribution of posteriori model and $p(X)$ is boundary likelihood which is solved through the following equation:

$$p(X) = \int p(X|\theta)P(\theta)\, d\theta$$

Firstly, target position (x_d, y_d) is randomly selected in new training region as input of GP model, and its output is predicted joint angle $\boldsymbol{\mu}_{\boldsymbol{a}}^{*}$. Through posture evaluation and mechanism of adjustment, actual position (x_a, y_a) of gripper is distributed nearby the target position, but it will not necessarily coincide with target position. Actual position (x_a, y_a) and corresponding joint angle parameter are added into training set as new samples. After enough samples in new training region are obtained, posteriori Gaussian distribution containing new training region will be established. The grasping evaluation mechanism is established according to actual grasping location and joint angle parameters generated through posteriori model. The termination conditions of posteriori model are defined. The evaluation function is designed as below:

$$r = cV_r e^{-\lambda \Delta d} + V_r^{'}, \Delta d = \sqrt{(x_a - x_d)^2 + (y_a - y_d)^2}$$

where c represents whether manipulator goes through posture adjustment, c = 0 means yes and c = 1 means no, and extra bonus will be obtained when the robot doesn't need posture adjustment. V_r is the reward value when deviation Δd is 0, λ is a sufficiently large parameter which can guarantee that $\mathrm{e}^{-\lambda \Delta d}$ converges to 0 when Δd is great enough. $V_r^{'}$ is the reward value for a newly added sample.

Termination condition of model training is as below:

When $R = \sum_{i=1}^{n} r_i > \kappa$ is satisfied, model training is terminated.

The self-learning algorithm is shown in following as Algorithm 1.

Algorithm 1

Input: sample set of demonstration: $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$; observation space $\boldsymbol{O}$;
cumulative reward R = 0; distribution of prediction: $\boldsymbol{h} = [\boldsymbol{a}, \boldsymbol{o}]^T \sim (\boldsymbol{\mu}, \boldsymbol{K})$
for t = 1,2,… do
  $\forall \boldsymbol{o} \in \boldsymbol{O}: \boldsymbol{a} = \boldsymbol{\mu}_a + \boldsymbol{K}_{ao}(\boldsymbol{K}_{oo} + \sigma_n^2 \boldsymbol{I})^{-1}(\boldsymbol{o}^* - \boldsymbol{\mu}_o)$;
  Loop A: $(\boldsymbol{\theta}_z, \boldsymbol{P}) = \mathbf{T}(\boldsymbol{a})$
    if $\forall \mathrm{i} \in \{1,2,3\}: |\theta_{zi}| < \delta, then$
      end Loop A
    else
      $\boldsymbol{a} = \boldsymbol{a} + \boldsymbol{w}\boldsymbol{\theta}_z^T$
  Loop B: $(\boldsymbol{\theta}_z, \boldsymbol{P}) = \mathrm{T}(\boldsymbol{a})$
    if $|\boldsymbol{P}| < h$ and $\exists \mathrm{i} \in \{1,2,3\}: |\theta_{zi}| > \delta, then$
      Goto Loop A
    else if $|\boldsymbol{P}| \geq \mathrm{h}$, then
      $\boldsymbol{a} = \boldsymbol{a} + \boldsymbol{w}'\boldsymbol{P}$
    else
      end Loop B
  execute a, obtain o′ in $\boldsymbol{O}$, then add $\mathbf{x}_{n+t} = (\boldsymbol{o}', \boldsymbol{a})^{\mathrm{T}}$ into $\boldsymbol{X}$;
  update the posterior distribution of $\boldsymbol{h}$: $\boldsymbol{p}(\boldsymbol{h}|\boldsymbol{X}, \boldsymbol{\theta}) = \boldsymbol{N}(\boldsymbol{\mu}, \boldsymbol{K} + \boldsymbol{\sigma}_n^2 \boldsymbol{I})$, **where**
  $\boldsymbol{\mu} = \frac{\sum_{i=1}^{n+t} x_i}{n+t}$, $(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}) = Cov\left(\left[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n+t}\right]^T\right)$;
  $r(\boldsymbol{o}, \boldsymbol{o}')$ = the reward of executing $\boldsymbol{a}$ in $\boldsymbol{O}$, R = R + r;
  if $\mathrm{R} = \sum_{\mathrm{i}=1}^{\mathrm{t}} \mathrm{r_i} > \kappa$
end for
Output: new sample set $\boldsymbol{X}$, new distribution of prediction $\boldsymbol{h}$

---

4 Simulation and Experiment

Simulation and experimental object is visual grasping platform based on UR3. UR3 contains 6 joint axes with high flexibility and operability. After new observational variable $\boldsymbol{o}^*$ is obtained, the trained GP model is used to calculate joint $\boldsymbol{a}^*$ which the robot needs to reach so as to realize robot self-adaptive grasping of the target object.

4.1 Simulation of Control Method

Gaussian self-adaptive grasping
Simulation uses Robotic Toolbox in MATLAB. Firstly, demonstration of the robot is performed, corresponding joint angle is input so that the gripper will grasp the object in vertical direction. Meanwhile, the certain grasping height is ensured so that gripper can grasp the target successfully. The robot joint angle and pixel coordinate of the gripper after coordinate transformation are recorded as training samples. They are input into the training sample set of GP model. In the simulation, 13 pixels' coordinates are uniformly collected in training region, and training data are shown in Table 1. After training samples are obtained, maximum likelihood estimations of mean value vector $\boldsymbol{\mu}$ and covariance matrix $\left(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}\right)$ of GP are

obtained through Eqs. (6) and (7). New observation and robot joint comply with the same probability distribution. When there is new observation input $\boldsymbol{o}^{*}$, joint angle $\boldsymbol{\mu}_{a}^{*}$ which allows the robot complete grasping at maximum probability can be solved through Eq. (9).

Distribution of training sample points and test sample points inside pixel plane is as shown in Fig. 5. The left green dotted box expresses the training region and the right red dotted box expresses untrained one. Blue points are training samples uniformly collected, and the other points are test samples. The results of grasping are judged according two rules. First, the distance from gripper and target location should be small enough. When the distance is smaller than 10 mm, it can be considered as a successful grasp. Second, height of gripper should be appropriate. In the simulation, when height of gripper is equal to 69 ± 4 mm, it can be considered that grasping is successful. Based on the two rules, the green points in the figure are successful samples. The yellow points express samples with too large deviation of grasping distance. The red points express samples with inappropriate grasping height.

As shown in Fig. 5, most test points in the training region can be successfully grasped. It can be seen that under small sample size, Gaussian process model has considerable performance in the training region. However, test points nearby boundary of training region and those beyond training region almost failed to grasp, and grasping effect is unsatisfying. Therefore, Gaussian process model relies on priori samples and has poor performance under the circumstance in which there are no priori samples.

Table 1 Training data from demonstration

Num	Observations		Corresponding joint angle (rad)					
	Pixel x	Pixel y	Base	Shoulder	Elbow	Wrist 1	Wrist 2	Wrist 3
1	239	88	−0.134	−1.730	−2.022	−0.965	1.573	0.007
2	347	88	0.003	−1.661	−2.103	−0.951	1.574	0.145
3	455	87	0.149	−1.609	−2.161	−0.944	1.575	0.291
4	455	191	0.130	−1.768	−1.976	−0.970	1.574	0.271
5	347	192	0.003	−1.810	−1.920	−0.985	1.573	0.145
6	239	192	−0.118	−1.868	−1.841	−1.008	1.572	0.023
7	239	297	−0.105	−2.004	−1.639	−1.073	1.571	0.036
8	347	297	0.003	−1.953	−1.717	−1.045	1.572	0.144
9	454	296	0.116	−1.917	−1.772	−1.026	1.573	0.257
10	401	244	0.062	−1.861	−1.852	−1.003	1.573	0.204
11	293	245	−0.055	−1.907	−1.785	−1.024	1.572	0.086
12	293	140	−0.062	−1.767	−1.977	−0.973	1.573	0.079
13	401	140	0.071	−1.712	−2.045	−0.958	1.574	0.212

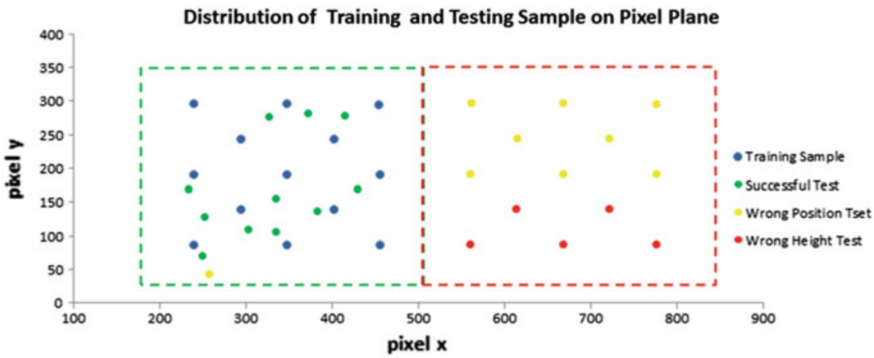

Fig. 5 Distribution of training and testing samples on pixel plane

Self-learning grasping

It's assumed that in new training region, probability distribution which observations and robot joint angles subject to is identical with that in the previous one. The trained GP model is taken as priori model, 12 test sample points are uniformly collected in new training region. The obtained joint angle and grasping errors are shown in Table 2.

Table 2 shows that in new training region, all test points failed to grasp. The average grasping error is 25.4 mm and maximum deviation reaches as high as 49 mm. Grasping height of more than 1/3 test points do not meet requirement. Then, self-learning is implemented in the new training region, and steps are as below:

Table 2 Test samples in new training region

No.	Pixel *x*	Pixel *y*	Height	Error	Base	Shoulder	Elbow	Wrist 1	Wrist 2	Wrist 3
1	560	88	63.3	15.26	0.253	−1.568	−2.232	−0.912	1.576	0.395
2	668	88	60.2	29.77	0.378	−1.518	−2.299	−0.895	1.577	0.521
3	777	87	56.5	49.08	0.505	−1.465	−2.369	−0.877	1.578	0.647
4	776	191	65.7	39.40	0.503	−1.610	−2.176	−0.924	1.577	0.645
5	668	192	67.8	21.63	0.378	−1.662	−2.107	−0.942	1.576	0.520
6	560	192	69.2	8.86	0.252	−1.713	−2.041	−0.960	1.575	0.394
7	561	297	70.5	11.38	0.253	−1.859	−1.847	−1.007	1.574	0.394
8	668	297	70.7	22.44	0.377	−1.809	−1.914	−0.990	1.575	0.519
9	776	296	70.4	38.83	0.503	−1.757	−1.982	−0.972	1.576	0.644
10	722	244	69.3	28.64	0.440	−1.709	−2.045	−0.957	1.576	0.582
11	615	245	70.2	14.02	0.316	−1.761	−1.977	−0.975	1.575	0.458
12	614	140	65.8	17.57	0.315	−1.615	−2.170	−0.927	1.576	0.457

1. Test samples are selected in new training region, and robot joint angles are obtained according to previous GP model.
2. In simulation, forward kinematic solution is solved according to joint angles. Iterative loop will be performed and joint angles will be adjusted if it's judged the posture or height is inappropriate. Then, go to the next step.
3. Corresponding pixel coordinates of the gripper are observed and recorded. which will be taken as new samples together with newly obtained robot joint angles. Input new samples into the training set of GP so as to update the whole distribution.
4. Reward value is calculated and updated, and it's judged whether self-learning training is completed. If it's judged that it's not completed, then return to 1, or otherwise, end the loop.

During learning process, the generated new samples are input into training set of GP so as to update the whole distribution. As shown in Fig. 6, the first row is distribution of predicted values of 6 joint angles in new training region when new training samples are not added. The second row is distribution after 6 new samples are added. The third row is distribution after 12 new samples are added. With increase in the number of new samples, distribution of robot joint angles changes. Reward value reaches threshold value through test sample point collection and updating of posteriori model for 12 times so that self-learning training is completed.

After self-learning of grasping, 100 sample points ate randomly tested in the new region, and distribution of test results is shown in Fig. 7. In the new training region, 83% test sample points can be successfully grasped. 14% test sample points failed to grasp because of deviation of position, where distance errors of 11% samples are within 10 ~ 15 mm. 3% samples failed to grasp because their grasping height is wrong. On the whole, posteriori GP model after self-learning has favorable effect in

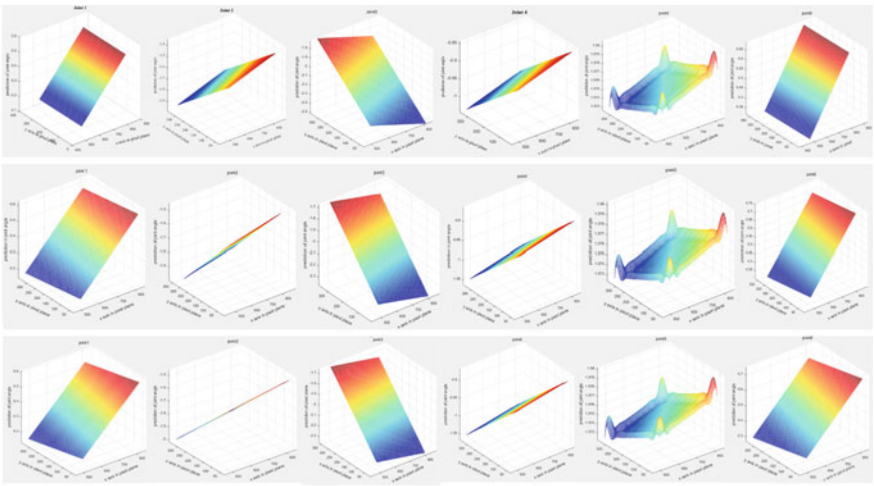

Fig. 6 Variation of the distribution of manipulator's joints

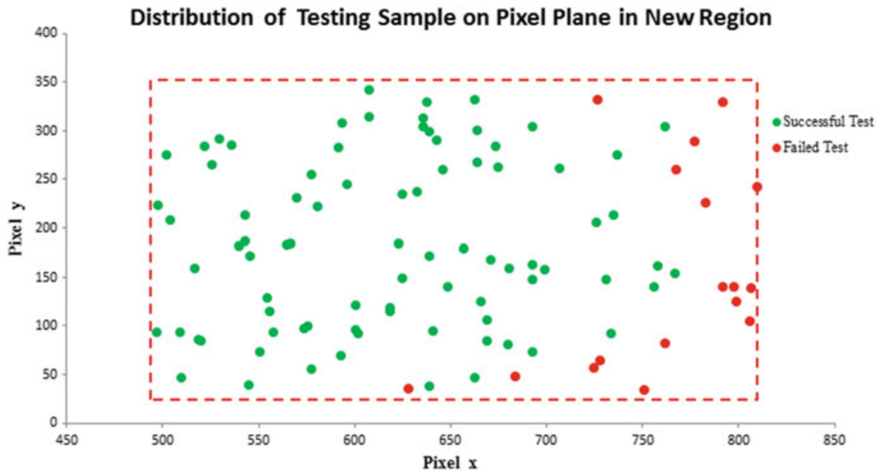

Fig. 7 Distribution of test samples in new region after self-learning

the new training model. Furthermore, most fail samples are distributed far away from training sample points. This conforms to characteristics of GP model. In the region with concentrated samples, the prediction performance is good, and in the region with sparse samples or being far away from samples, the performance of prediction tends to be poor.

4.2 Experiment

The experimental platform is shown in Fig. 8a, the camera is placed above the platform. Target object is a colored block on the table. UR3 is selected as manipulator and is arranged at left side of the platform. SRT soft gripper is selected as end-effector. The soft gripper is of adaptivity to the shape of grasping object, which improves grasping success rate and contributes to training effect. Camera selected in the experiment is MV-EM200C/M and resolution is 1600 × 1200.

In the experiment, firstly GP model is trained. Profile extraction and calculation of central point of the target object are implemented through industrial camera, image processing interface is as shown in Fig. 8b. The pixel coordinates of central position of target object (green wood block) are obtained as observations $\boldsymbol{o}(x, y)$.

Demonstration is implemented through manual operation, robot joint angles are adjusted, the end-effector is set to approach the target object in reasonable posture, robot joint angle $\boldsymbol{a}$ is recorded when successful grasping is ensured, and observations and joint angles are taken as training samples of GP model. Similar to simulation, 13 sample points are collected as training samples, and then maximum likelihood

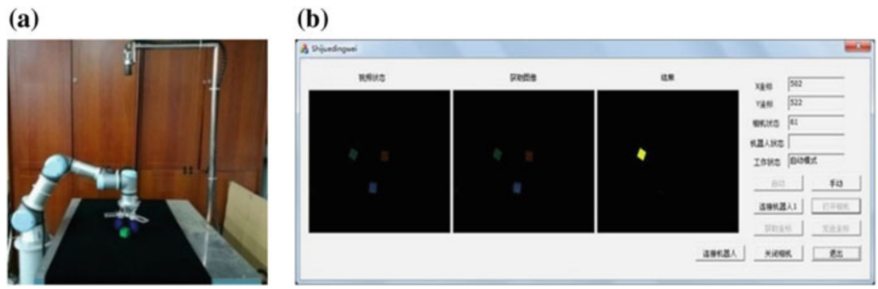

Fig. 8 Experiment of grasping task: **a** experimental platform; **b** acquiring observation variables of target object

estimation values of mean value vector $\boldsymbol{\mu}$ and covariance matrix $\left(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}\right)$ of GP are obtained through Eqs. (6) and (7). Training data are shown in Table 3.

Green wood blocks are randomly placed in training region, and 12 samples points are tested. The distribution of training samples and test samples are shown in Fig. 9. Blue points are training samples which are uniformly collected in training region. Green points are successful samples and red points are failed samples. Similar to the simulation, most test samples can be successfully grasped, several failed samples are located at edges of training region. In general, in the training region, self-adaptive grasping of GP has decent performance result. However, similar to simulation, beyond the training region, previous GP model is tried for estimation to generate joint angles result, and nearly all test samples failed to grasp. GP model has poor performance in the region without prior knowledge.

Table 3 Training data from demonstration

No.	Observations		Corresponding joint angle (rad)					
	Pixel x	Pixel y	Base	Shoulder	Elbow	Wrist 1	Wrist 2	Wrist 3
1	240	455	−0.092	−2.220	−1.239	−1.271	1.573	6.188
2	346	455	0.001	−2.171	−1.324	−1.233	1.575	6.281
3	453	454	0.096	−2.136	−1.382	−1.208	1.577	0.093
4	453	559	0.088	−2.304	−1.092	−1.331	1.576	0.085
5	347	559	0.002	−2.340	−1.026	−1.362	1.574	6.281
6	240	559	−0.083	−2.395	−0.925	−1.410	1.572	6.196
7	241	667	−0.076	−2.650	−0.455	−1.620	1.571	6.204
8	347	670	0.002	−2.564	−0.620	−1.530	1.573	6.282
9	454	663	0.082	−2.514	−0.701	−1.513	1.575	0.078
10	400	611	0.043	−2.418	−0.883	−1.427	1.574	0.040
11	294	612	−0.039	−2.469	−0.787	−1.474	1.573	6.240
12	293	507	−0.043	−2.276	−1.142	−1.312	1.574	6.236
13	400	507	0.047	−2.233	−1.218	−1.277	1.576	0.043

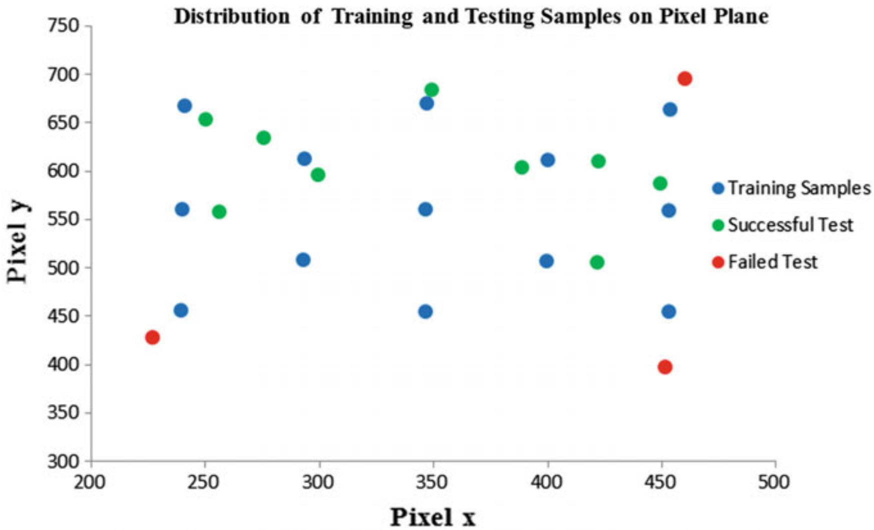

Fig. 9 Distribution of training and testing samples on pixel plane

Self-learning training is carried out according to simulation steps, subsequently, green blocks are randomly placed in new training region, and 250 samples are tested, and test results are shown in Table 4. Among 250 experimental tests, the number of successful grasping is 194 occupying 77.6% which is slightly lower than in simulation. That is possibly caused by errors of camera and precision of manipulator. Among unsuccessful samples, samples with deviation of grasping location occupy 12.4% which constitute the main type of unsuccessful grasping. Blocks are successfully grasped but drop during movement process of manipulator, it belongs to unsafe grasping situation, and this occupies 6.4%. The situation of nothing grasping or gripper touching experimental platform due to wrong grasping height occupies 3.6%. In general, after self-learning, posteriori GP has favorable performance result in the new training region.

Table 4 Grasping test in experiment

The result of grasping	No.	Proportion
Successful grasping	194	77.6%
Grasp wrong position	31	12.4%
Unstable	16	6.4%
Grasp wrong height	9	3.6%

5 Conclusion

Robot self-adaptive grasping strategy based on Gaussian process was proposed. The pose and position information of the target object obtained through visual grasping and corresponding robot joint variables were associated. On the condition that only a small sample size was needed, the robot was made to learn from artificial demonstration samples. The robot self-learning grasping method based on Gaussian process and Bayesian algorithm was presented. The robot could use previous Gaussian process model as priori model to implement self-learning grasping in the region without prior knowledge, training scope of Gaussian process model was expanded and adaptability of robot grasping was improved. This method omits the complex and time-consuming calibration of visual system. Besides, choosing similar results with the demonstration samples, it does not need to solve the inverse kinematic and the optimal solution which improves the efficiency of grasping. When grasping environment changes, the previous learning experience can be used as priori knowledge. The self-learning is able to complete the grasping task in new environment in relative high success rate without repeating calibration process, which reduces the workload of operators.

References

1. Kroemer OB, Detry R, Piater J et al (2010) Combining active learning and reactive control for robot grasping. Robot Auton Syst 58(9):1105–1116
2. Lenz I, Lee H, Saxena A (2013) Deep learning for detecting robotic grasps. Int J Robot Res 34(4–5):705–724
3. Manti M, Hassan T, Passetti G et al (2015) A bioinspired soft robotic gripper for adaptable and effective grasping. Soft Robot 2(3):107–116
4. Levine S, Finn C, Darrell T et al (2016) End-to-end training of deep visuomotor policies. J Mach Learn Res 17(39):1–40
5. Zeng A et al (2017) Multi-view self-supervised deep learning for 6D pose estimation in the amazon picking challenge. In: Proceedings of the IEEE international conference on robotics and automation, IEEE, Singapore, pp 1386–1383
6. Garcia-Sillas D et al (2016) Learning from demonstration with Gaussian processes. In: IEEE conference on mechatronics, adaptive and intelligent systems, IEEE, Hermosillo, pp 1–6
7. Shimojo M, Namiki A, Ishikawa M et al (2004) A tactile sensor sheet using pressure conductive rubber with electrical-wires stitched method. IEEE Sens J 4(5):589–596
8. Bekiroglu Y, Laaksonen J, Jorgensen JA et al (2011) Assessing grasp stability based on learning and haptic data. IEEE Trans Rob 27(3):616–629
9. Dang H, Allen PK (2014) Stable grasping under pose uncertainty using tactile feedback. Auton Robots 36(4):309–330
10. Chebotar Y et al (2016) Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning. In: IEEE/RSJ international conference on intelligent robots and systems, IEEE, Daejeon, pp 1960–1966
11. Haschke R et al (2005) Task-oriented quality measures for dextrous grasping. IEEE international symposium on computational intelligence in robotics and automation. IEEE, Espoo, pp 689–694

12. Roa MA, Suárez R (2015) Grasp quality measures: review and performance. Auton Robots 38 (1):65–88
13. Miller AT et al (2003) Automatic grasp planning using shape primitives. In: Proceedings of the IEEE international conference on robotics and automation, vol 2, IEEE, pp 1824–1829
14. Ciocarlie MT, Allen PK (2009) Hand posture subspaces for dexterous robotic grasping. Int J Robot Res 28(7):851–867
15. Li Y, Saut J, Pettré J et al (2015) Fast grasp planning using cord geometry. IEEE Trans Rob 31(6):1393–1403
16. Zhang Z (2000) A flexible new technique for camera calibration. IEEE Trans Pattern Anal Mach Intell 22(11):1330–1334
17. Wang Y, Liu CJ, Ren YJ et al (2009) Global calibration of visual inspection system based on universal robots. Opt Precis Eng 17(12):3028–3033
18. Tsai RY, Lenz RK (1989) A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. IEEE Trans Robot Autom 5(3):345–358
19. Levine S et al (2016) Learning hand-eye coordination for robotic grasping with largescale data collection. International symposium on experimental robotics. Springer, Berlin, pp 173–184
20. Pinto L et al (2016) Supersizing self-supervision: learning to grasp from 50 k tries and 700 robot hours. In: Proceedings of the IEEE international conference on robotics and automation, IEEE, Stockholm, pp 3406–3413
21. Finn C et al (2017) Deep visual foresight for planning robot motion. In: Proceedings of the IEEE international conference on robotics and automation, IEEE, Singapore, pp 2786–2793
22. Hutchinson S, Hager GD, Corke PI (1996) A tutorial on visual servo control. IEEE Trans Robot Autom 12(5):651–670
23. Chaumette F, Hutchinson S (2006) Visual servo control. I. basic approaches. IEEE Robot Autom Mag 13(4):82–90
24. Siradjuddin I et al (2012) A position based visual tracking system for a 7 DOF robot manipulator using a kinect camera. The 2012 international joint conference on neural networks. IEEE, Brisbane, pp 1–7
25. Thomas J et al (2014) Toward image based visual servoing for aerial grasping and perching. In: Proceedings of the IEEE international conference on robotics and automation, IEEE, Hong Kong, pp 2113–2118
26. Wang Y, Lang H, Silva CW (2010) A hybrid visual servo controller for robust grasping by wheeled mobile robots. IEEE/ASME Trans Mechatron 15(5):757–769
27. Lin Y et al (2014) Grasp planning based on strategy extracted from demonstration. IEEE/RSJ international conference on intelligent robots and systems. IEEE, Chicago, pp 4458–4463
28. Sauser EL, Argall BD, Metta G et al (2012) Iterative learning of grasp adaptation through human corrections. Robot Auton Syst 60(1):55–71
29. Faria DR, Martins R, Lobo J et al (2012) Extracting data from human manipulation of objects towards improving autonomous robotic grasping. Robot Auton Syst 60(3):396–410
30. Rasmussen CE, Williams CK (2004) Gaussian processes in machine learning. Lect Notes Comput Sci 3176:63–71
31. Ghadirzadeh A et al (2016) A sensorimotor reinforcement learning framework for physical human-robot interaction. IEEE/RSJ international conference on intelligent robots and systems. IEEE, Daejeon, pp 2682–2688
32. Ghadirzadeh A et al (2014) Learning visual forward models to compensate for self-induced image motion. The 23rd IEEE international symposium on robot and human interactive communication. IEEE, Edinburgh, pp 1110–1115