2009 Ninth International Conference on Intelligent Systems Design and Applications

Acquisition of image feature on collision for robot motion generation

Taichi Okamoto and Yuichi Kobayashi*
Tokyo University of Agriculture and Technology / RIKEN RTC*
Department of Electrical and Electronic Engineering
2-24-16 Nakacho, Koganei, Tokyo, JAPAN
yu-koba@cc.tuat.ac.jp

Masaki Onishi
National Institute of Advanced Industrial Science and Technology (AIST)
Information Technology Research Institute
1-1-1 Umezono, Tsukuba, Ibaraki, JAPAN

Abstract

It is important for robots that act in human-centered environments to build image processing in a bottom-up manner. This paper proposes a method to autonomously acquire image feature extraction that is suitable for motion generation while moving in unknown environment. The proposed method extracts low level features without specifying image processing for robot body and obstacles. The position of body is acquired in image by clustering of SIFT features with motion information and state transition model is generated. Based on a learning model of adaptive addition of state transition model, collision relevant features are detected. Features that emerge when the robot can not move are acquired as collision relevant features. The proposed framework is evaluated with real images of the manipulator and an obstacle in obstacle avoidance.

1. Introduction

In building a robot that act in human-centered environments, it is not practical to embed the information of all possible objects to be recognized. Thus, it is important that robots acquire information autonomously from unknown environment. An important issue is development of image processing ability. Minato et al. pointed out that it is important for the development of intelligent robot that the robot can generate feature extractor of the image autonomously [1]. They proposed to obtain image processing filters autonomously based on maximization of conditional entropy defined by image inputs and actions, where desired action for each image input is given as teaching signals. The learning ability of robot would be improved if motion generation is concurrently considered in feature extraction learning.

In the case where a robot does not have any information on image input, to discover the position of its own body in the image is an important issue. Fitzpatrick et al. showed a bottom-up acquisition of body and object representation [2]. They proposed to find the robot body as what moves in the image first when motor command is given and then find an object as what moves the next to its body. The robot body and the object were acquired by difference processing. Stoytchev proposed a method to discover the robot body in the image and acquire similarity transformation between images with different scales [3]. In this research, markers that move concurrently with motor commands are detected as a part of body. From the viewpoint of autonomous image processing, however, it is specialized approach for extraction of robot arm due to using markers. Though acquisition of body image has been actively investigated, its application to motion generation including state transition prediction model, planning method and controller has not been sufficiently discussed. If robot can learn lower level image features in a bottom-up manner instead of using difference processing, the applicability of the robot can be broadened.

This paper proposes an acquisition of low level image feature extraction which is closely coupled with motion generation via state transition estimation. In this paper, collision avoidance by a two-link manipulator is implemented as a task for the robot system. We use SIFT (Scale Invariant Feature Transform) features to find robot body image and detect collision relevant features. SIFT is one of the methods that describe features in image proposed by Lowe [5]. SIFT can describe constant features for rotation and variation of scales and has been applied to robot vision recently (e.g. [6]).

An advantage of the proposed method is that it does not depend on specific hand-coded design. Thus, it can be eas-

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.113

ily applied to different manipulators. Robot can acquire action which suits environment autonomously thorough trial and error without advance informations of the environment and the robot. Our approach generates the motion of the robot with reinforcement learning [7] in image coordinate instead of using configuration space [8]. The reason for this is that it is easy to control the robot body when a target position of the body is given in the image coordinate. Another advantage is that the proposed framework can be regarded as an extension of MOSAIC (Modular Selection and Identification for Control)[4], which is known as controller for nonlinear and unsteady systems with multiple- modules structure.

The experimental setup is described in section 2. Section 3 describes the proposed method. The verification of the method by experiment is given in section 4. Finally, section 5 concludes this paper.

2 Problem Setting

Suppose there is a manipulator where kinematic parameters are unknown. There are two tasks for the robot system. The first task is to move a part of the body to a target position in the 2D image coordinate, where representation of the body (which part is the body in the image) is not given in advance. The second task is to move the part of the body to a target position while avoiding collision with obstacles. Here, representation of the obstacles is not given in advance, either.

3 Image Feature Acquisition with Collision Avoidance

First, the robot builds representation of its own body in the image obtained by the camera. The task for the robot is to control the center of the body relevant features (hereafter called the position of the body) to a target position. The position of the manipulator is discovered using correspondence between a motor command and motion of features in the image. This body detection does not utilize any specific knowledge on the image of the robot manipulator except for synchronousness between motion of the manipulator in the image and motor command. As low-level feature detection, we apply SIFT. In the first process, the manipulator can move freely without any collision with obstacles.

State transition models of the manipulator are generated by estimation of Jacobian using the position of the body and the displacement between corresponding features before and after motor command. By this, the robot can control to move the manipulator.

Next, the robot finds visual features of collision with obstacles for collision avoidance for motion generation. The process in this stage deals with the collision between the manipulator and the obstacle.

Finally, the robot moves while avoiding the obstacle, after acquiring the feature that is relevant to collision. Dynamic programming with updated dynamics models is used in motion generation.

3.1 Extraction of body relevant features

Features which move in the image when the manipulator is commanded to move are extracted as body relevant features. In this subsection it is assumed that there is no collision with obstacles or no objects that moves irrelevantly to the manipulator. First, SIFT features are calculated for two images, one before motor command (displacement of joint angle $\Delta \boldsymbol{q}$) and the other after the command. SIFT features (keypoints) contain 128 dimensional feature vectors and two dimensional position vectors in the image coordinate. Using the feature vectors, matching is calculated (by comparing Euclidian norms) among keypoints in the two images.

Suppose n keypoints have been matched in the two images. Let $\boldsymbol{p}_i(t)\ (i = 1, 2, \cdots, n) \in \mathbb{R}^2$ denote the position of i-th matched feature in the image before the motor command and $\boldsymbol{p}_i(t + \Delta t) \in \mathbb{R}^2$ denote the position of i-th matched feature which corresponds to $\boldsymbol{p}_i(t)$ in the image after the command. The displacement of i-th feature in the image coordinate is defined by

$$\boldsymbol{v}_i = \boldsymbol{p}_i(t + \Delta t) - \boldsymbol{p}_i(t), \quad i = 1, 2, \cdots, n. \tag{1}$$

To find a group of keypoints that move similar directions with similar positions, clustering with $[\boldsymbol{v}_i^T, \boldsymbol{p}_i^T]^T \in \mathbb{R}^4$ is calculated. Mean shift [9] is used for the clustering.

The cluster which has the largest mean displacement is regarded as the body relevant features since the displacement of body relevant features is supposed to have the largest value. The cluster for the body relevant feature is decided as

$$j^* = \underset{j}{\operatorname{argmax}} \frac{1}{|C_j|} \sum_{i \in C_j} \| \boldsymbol{v}_i \|, \tag{2}$$

where $j = 1, 2, \cdots, m$ denotes the index of cluster and C_j denotes the set of j-th cluster. Figure 1 shows an outline of acquisition of body relevant features. Blue circles in the figure indicate SIFT features (keypoints). The size of circle shows scale of the feature. The red ellipse indicates the cluster of keypoints with similar displacements and positions. Figure 2 shows an example of matching between SIFT features. In the figure, blue circles indicate features that are matched and red ones indicate mismatched.

3.2 Generation of state transition models

In the second step, the relation between motor command $\Delta \boldsymbol{q}$ and displacement of the position of the body is estimated. This relation can be used to predict the motions of

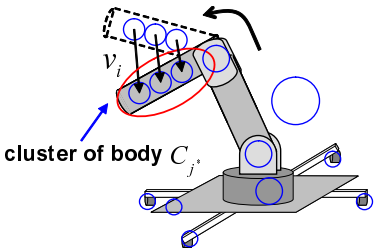
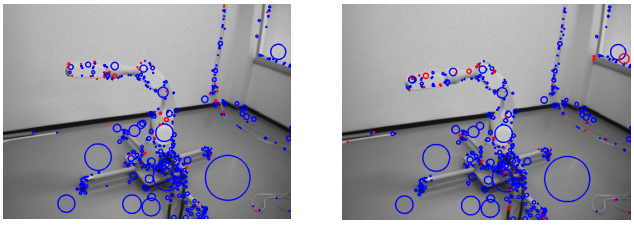


Figure 1. Outline of clustering to extract body relevant features



(a) before motor command (b) after motor command

Figure 2. Example of matching of SIFT features

body relevant features and to realize desired motions of the features. In general, the relation between displacement of joint angle and velocity of the manipulator in the image is represented as follows.

$$\dot{\boldsymbol{x}} = \boldsymbol{J}_q \dot{\boldsymbol{q}}, \tag{3}$$

where $\dot{\boldsymbol{x}}$ denotes the velocity of the hand in the image coordinate, $\dot{\boldsymbol{q}}$ denotes the angular velocity of joint, and $\boldsymbol{J}_q$ denotes Jacobian.

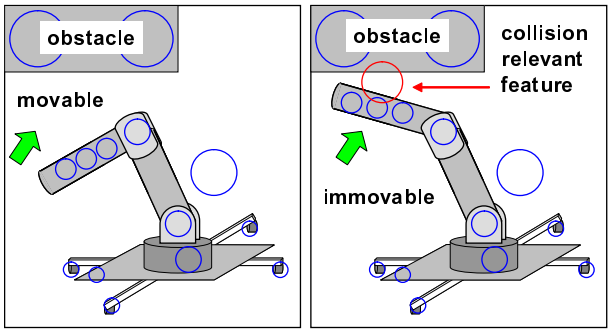
Suppose l small displacements of joint angle $\Delta\boldsymbol{q}_1, \Delta\boldsymbol{q}_2, \cdots, \Delta\boldsymbol{q}_l \in \mathbb{R}^2$ are tested as motor commands at the same initial angle of $\boldsymbol{q}$. Let $\boldsymbol{Q} = [\Delta\boldsymbol{q}_1, \Delta\boldsymbol{q}_2, \cdots, \Delta\boldsymbol{q}_l]$ and corresponding displacements of body relevant features in the image coordinate be defined as $\boldsymbol{X} = [\Delta\boldsymbol{x}_1, \Delta\boldsymbol{x}_2, \cdots, \Delta\boldsymbol{x}_l]$. It holds from $\boldsymbol{X} = \boldsymbol{J}_q\boldsymbol{Q}$ that the least square approximation of $\boldsymbol{J}_q$ is given by

$$\boldsymbol{J}_q = \boldsymbol{X}(\boldsymbol{Q}^T\boldsymbol{Q})^{-1}\boldsymbol{Q}^T. \tag{4}$$

$\boldsymbol{J}_q$ is approximated for pairs of (q_1, q_2). $q_1 - q_2$ space is discretized into $n_1 \times n_2$ grids and $\boldsymbol{J}_q$ is approximated at each grid.

3.3 Acquisition of collision relevant features

In the third step, features that are relevant to collisions between the manipulator and obstacles are acquired. Here, the motor command $\Delta\boldsymbol{q} \in \mathbb{R}^2$ is given to the robot randomly while the workspace contains an obstacle. Using the



(a) free (b) collision

Figure 3. Outline of acquisition of collision relevant features

state transition model described in the previous section, the position of a body relevant feature at the next time step can be predicted as

$$\hat{\boldsymbol{x}}(t + \Delta t) = \boldsymbol{x}(t) + \boldsymbol{J}_q \Delta\boldsymbol{q}(t). \tag{5}$$

When the robot collides with the obstacle, the actual motion of the manipulator does not match the prediction of Eqn. (5). That is, the actual state transition will be

$$\boldsymbol{x}(t + \Delta t) = \boldsymbol{x}(t). \tag{6}$$

In such case, a new prediction model is created to account for the different state transition. Features that emerge in this new transition for the first time are memorized for switching the state transition models. The idea of finding collision relevant features is depicted in Figure 3. In Figure 3(a), the robot can move the link to the upper direction. This small motion matches the prediction given by the state transition model of Eqn. (5). On the other hand, when the link comes close to the obstacle as shown in Figure 3(b), the link can not be moved to the same direction any more. The first model of Eqn. (5) can not predict the transition but the new model of Eqn. (6) can account for this case. The red circle indicates a feature that emerged at this situation. This feature can be used to switch the prediction models.

An example of detecting collision relevant features is shown in Figure 4. Red circles in Figure 4(a) indicate features that emerged when collision occurred, i.e., features that did not match features in the image of the previous time step.

Note here that the detected features for a collision mentioned above contains features that are actually not relevant to collision (see small red circles dispersed in Figure 4(b)). These irrelevant features are detected because of the process of matching of SIFT features under the influence of noise. To remove these irrelevant features, features that did

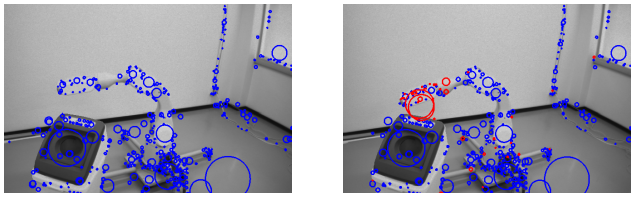


Figure 4. Example of acquisition of collision relevant features

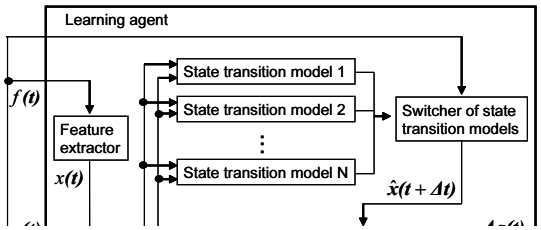


Figure 5. selection of state transition model

not match in the process described in 3.1 (red circles in Figure 2) are stored in advance. By comparing the detected features with the stored mismatching features, the collision relevant features can be extracted.

The flow of processing for motion generation is depicted in Figure 5, where $\boldsymbol{f}$ denotes SIFT feature vectors. The position of the body part $\boldsymbol{x}$ is extracted from $\boldsymbol{f}$ and fed into planner/controller and each state transition model. The output of state transition models are selected by the switcher and utilized to modify planning and control. In our application, two state transition models described by Eqn. (5) and Eqn. (6) are obtained. When collision relevant features are detected, the transition model is switched from model Eqn. (5) to model Eqn. (6).

3.4 Motion generation

Dynamic programming, which is a sort of reinforcement learning, is used for motion generation with adaptive modification of state transition probabilities. The state for planning and control is the position of the body in the 2D image coordinate. The state is discretized into $m_1 \times m_2$ grids. Actions are defined as transitions to the neighbor state grids to four directions (up, down, right and left). Update of the value functions at each grid (each state s and action a) is given as

$$V(s) = \max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')] \quad (7)$$

$$Q(s,a) = \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]. \quad (8)$$

$V(s)$ and $Q(s,a)$ denote the state value function and the action value function, respectively. a is action. $P^a_{ss'}$ denotes the state transition probability and the state transition is assumed be controlled deterministically in this application. γ denotes the discount factor and R denotes the reward. Initially, the value functions are calculated with $R = 0$ for the target position and $R = -1$ for all other states and actions.

When an action is decided by the action value function at each state, the motor command is calculated by $\Delta \boldsymbol{q} = \boldsymbol{J}_q^{-1} \Delta \boldsymbol{x}$, where $\Delta \boldsymbol{x}$ is given by the action on the state grid.

If there are collision relevant features, the state transition model is switched to the model which gives prediction of Eqn. (6). The value iteration of Eqn. (7) is processed with the new transition model at the colliding position and it results in a trajectory that avoids the collision.

4 Experimental Results

4.1 Experimental Setup

We use two PCs as shown in Figure 6, the one controls the manipulator and the other gets images by a stereo camera. Two joints of the manipulator are controlled for reaching motion. The stereo camera is used to get 2D image and range (3D) image, which is calculated by hardware of the stereo vision system. 2D and 3D images are stored with various postures of the manipulator off-line. With those images, a simulation with real images of the manipulator is built. In the simulation, the input image of the manipulator can be obtained when a motor command (small displacement of joint angles) is given. Besides, 3D images of obstacles are also taken in advance. By calculating interference between 3D pixels of the manipulator and the obstacles, collision can be simulated. Thus, the manipulator does not move when collision with obstacles occurs in the simulation.

4.2 Acquisition of body relevant features

In Figure 7, the results of clustering with mean shift using position and displacement of features which are detected by SIFT matching are shown. It can be seen that some keypoints on the manipulator are extracted in the two examples. Extraction of body is succeeded for all of 116 poses in the sense that the position of the body is around the manipulator link. More strictly evaluating, the position in 90.5% cases is on the link.

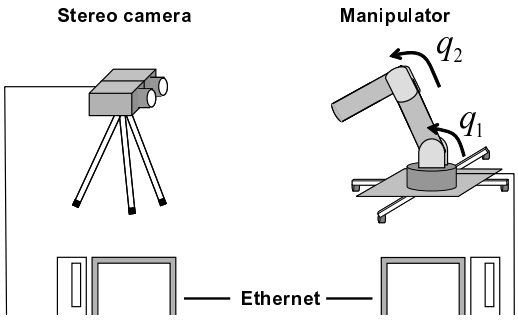


Figure 6. Experimental system

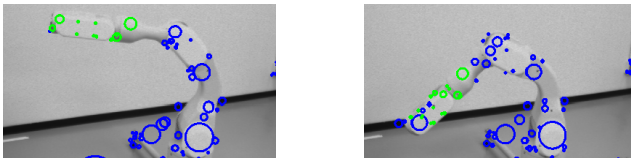


Figure 7. Example of acquisition of body relevant feature

Based on these results, Jacobian matrices are approximated with four displacements of joint angles for each grid, ± 5[deg] for q_1 and ± 10[deg] for q_2. The number of grids for the approximation is set as $n_1 = n_2 = 11$, within the range of $0 \leq q_1 \leq 50$[deg] and $50 \leq q_2 \leq 150$ [deg].

4.3 Acquisition of collision relevant features

To obtain images with collision with an obstacle, the robot moves its arm so that the arm contacts with the obstacle at various positions. Figure 8(b) shows all acquired collision relevant features.

Table 1 shows evaluation of collision detection. The result of recognizing whether given images contain collision relevant features or not. Totally 294 images are tested, including 19 images with collision relevant features.

4.4 Obstacle avoidance

The discretization of state space for the motion planning is set as $m_1 = m_2 = 10$. The range of discretization of

Table 1. Evaluation of collision detection

	Collision	[%]	No collision	[%]
Recognized as collision	15 / 21	[71.4]	4 / 273	[1.5]
Recognized as no collision	6 / 21	[28.6]	269 / 273	[98.5]

Figure 8. All acquired collision relevant features

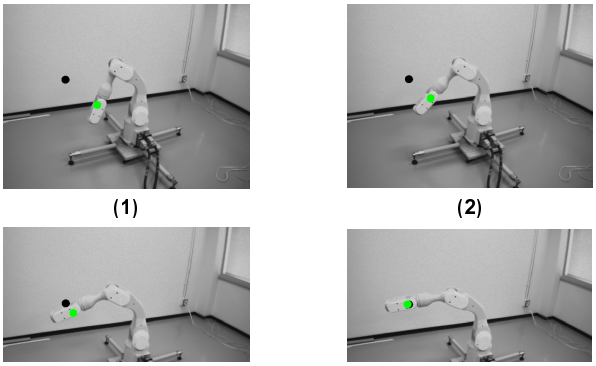


Figure 9. Result of reaching

state space in the image coordinate is $141.3 \leq x \leq 389.1$, $77.8 \leq y \leq 266.6$. Taking the effect of discretization into account, the task is judged to be finished when the distance between the position of the body and the target position is smaller than a threshold value. The discount factor is set as $\gamma = 0.9$. Figure 9 shows a trajectory of reaching motion realized by the proposed method. Green circles indicate the positions of the body and black circles indicate the target position.

Figure 10 shows a trajectory with an obstacle, where the initial configuration of the manipulator and the target position for the robot body is the same as the case shown in Figure 9. Red circles in the figure indicate collision relevant features. Note here that the position of the obstacle is different from the experiment described in 4.2. That is, the stored collision relevant features are tested in a similar but different situation to the learning phase. The difference between the positions of those two cases are 0.2[m] in the world coordinate. In comparison with Figure 9, where the manipulator moves straight toward the target, the manipulator once moved upward to avoid collision with the obstacle.

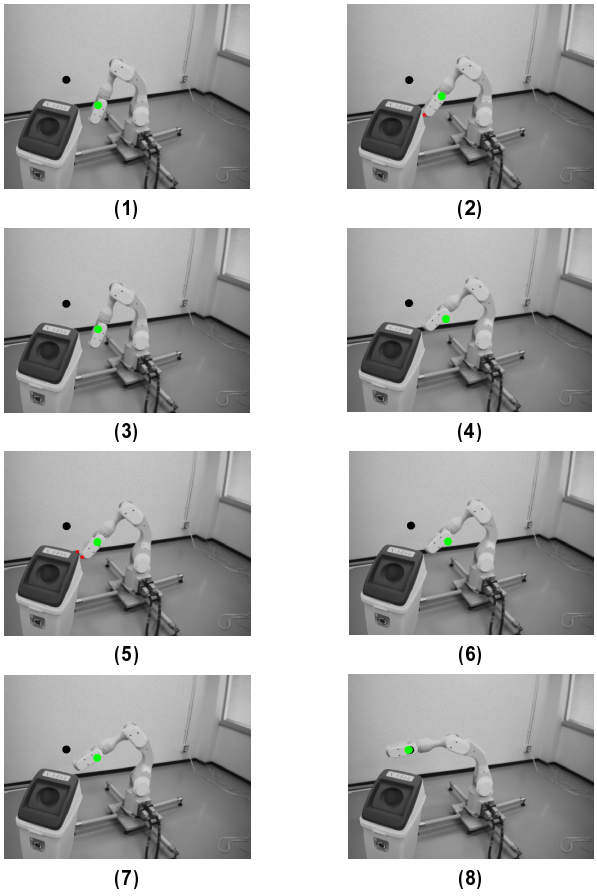


Figure 10. Result of avoidance obstacles

4.5 Discussion

Though a successful trajectory generation was shown in 4.3, there were failures for different conditions of initial configurations and target positions. Those failures were caused by two reasons: First, the clustering of the body relevant features was not consistent enough. The cluster with the largest displacement can vary depending on images, which can be also seen in Figure 9, 10 (green circles localize various parts of the arm). A possible solution for this is to utilize more information of SIFT features to consistently find the body part.

Second, generation of state transition model is not precise enough because of the SIFT matching error• ¶The result of average of 5 patterns of SIFT matching for the part of hand is shown Table.2. Failure 1 expresses there are no features to match. Failure 2 expresses matched feature is wrong. This result is not good enough increasing of matching accuracy is needed• •

5 Conclusion

In this paper, we proposed to acquire image feature extraction in a bottom-up manner that is suitable for motion generation of a manipulator. The proposed method first finds body relevant SIFT features autonomously, followed by acquisition of collision relevant SIFT features. Collision relevant features are extracted by the generation of the state transition models based on Jacobian approximation between motor command and body motion in the image. An appropriate collision avoidance behavior was realized with dynamic programming with on-line update of state transition model. One of our future works is to extend this idea of autonomous feature extraction to other kinds of robot motions, such as locomotion (localization) and manipulation of objects. The first step to such extensions will be to improve the generality of collision relevant features, because only one kind of obstacle was tested in the experiment.

Table 2. Success rate of SIFT matching

Feature	Ave. # of features	Percentage[%]
All	30.0	
Success	14.6	48.7
Failure 1	10.2	34.0
Failure 2	5.2	17.3

References

[1] T. Minato and M. Asada: "Towards Selective Attention: Generating Image Features by Learning a Visuo-Motor Map", Robotics and Autonomous Systems, vol.45, pp.211-221, 2003.

[2] P. Fitzpatrick, G. Metta, L. Natalc, S. Rao, G. Sandini: "Learning about objects through action - initial steps towards artificial cognition", Proc. of IEEE International Conference on Robotics and Automation, pp.3140–3145, 2003.

[3] A. Stoytchev: "Toward Video-Guided Robot Behaviors", Proceedings of the 7th International Conference on Epigenetic Robotics. pp. 165- 172. 2007.

[4] M. Haruno, D.M. Wolpert and M. Kawato: "Mosaic model for sensorimotor learning and control", Neural Computation, vol. 13, pp. 2201-2220, 2002.

[5] D. G. Lowe: "Object Recognition from Local Scale-Invariant Features", Proc. of IEEE International Conference on Computer Vision, pp.1150–1157, 1999.

[6] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann: "ARMAR-III: An integrated humanoid platform for sensory-motor control", in Proc. of the IEEERAS/ RSJ Int. Conf. on Humanoid Robots, pp.169–175, 2006.

[7] R. S. Sutton: "Reinforcement learning", MIT Press, 1998.

[8] J. C. Latombe: "Robot Motion Planning", Kluwer Academic Publishers, 1991.

[9] D. Comaniciu and P. Meer: "Mean Shift: A Robust Approach toward Feature Space Analysis", IEEE Trans. Pattern Anal. Machine Intell., vol. 24, no. 5, pp. 603-619, 2002.