

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Manifold-Based Robot Motion Generation

Yuichi Kobayashi

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.80973>

Abstract

In order to make an autonomous robot system more adaptive to human-centered environments, it is effective to let the robot collect sensor values by itself and build controller to reach a desired configuration autonomously. Multiple sensors are often available to estimate the state of the robot, but they contain two problems: (1) sensing ranges of each sensor might not overlap with each other and (2) sensor variable can contain redundancy against the original state space. Regarding the first problem, a local coordinate definition based on a sensor value and its extension to unobservable region is presented. This technique helps the robot to estimate the sensor variable outside of its observation range and to integrate regions of two sensors that do not overlap. For a solution to the second problem, a grid-based estimation of lower-dimensional subspace is presented. This estimation of manifold allows the robot to have a compact representation, and thus the proposed motion generation method can be applied to the redundant sensor system. In the case of image feature spaces with a high-dimensional sensor signal, a manifold estimation-based mapping, known as locally linear embedding (LLE), was applied to an estimation of distance between robot body and an obstacle.

Keywords: robot motion generation, redundant sensors, limited observation range, manifold by constraint

1. Introduction

Robotics is gathering attention for various applications such as autonomous navigation and manipulation of objects. It is highly expected that autonomous robots can act closer to humans, for example, in household environment. In reality, however, it is still very difficult to make those robots achieve various tasks in environments, which are not specifically structured for the robots. One of the reasons for this is that processes of recognition and

motion generation are all specifically designed according to individual specific cases. In unstructured environments, robots have to adapt to various changes of conditions in both recognition and motion control processes, which requires reconstruction of software by the human designers.

One possible approach to this problem, which might be promising but not straightforward, is to let the robot learn to build its representation for task execution based on its own experience (for example, discussed in the context of developmental robotics [1, 2]). In this approach, task-specific designs are omitted in recognition processes, which is quite different from the conventional robotics, where objects, robots, and environments are described by their coordinates (typically Cartesian) in world coordinate systems. For example, a mobile robot can achieve a navigation task based only on its information of distance sensors, while distance sensor information is normally converted to position of the robot based on its environmental map in simultaneous localization and mapping (SLAM) applications [3–5].

When we try to build a framework to allow an autonomous robot to build a state space for motion generation, the idea of manifold where only local coordinate systems are defined and relations among them are described is suitable for the purpose. The reason is that one kind of sensor does not provide thorough information about the robot system and its environment, and multiple sensors are often required to cover various situations, whereas relations among multiple sensor signals are not known in advance. Thus, an application of approximating manifold for robot motion generation is presented in this chapter.

First, an integration of multiple sensor spaces is presented. The proposed integration method is based on an idea that the system dynamics is continuous over a sensor signal space with respect to the control input. Redundant sensor signals are mapped onto a lower-dimensional subspace using a simple grid-based parameterization method, which was applied to a navigation problem of a mobile robot equipped with several distance sensors measuring distances to a wall. Second, an application of locally linear embedding (LLE) [6] to mapping from a high-dimensional image feature space to a low-dimensional space in robotic motion planning task is presented. No prior knowledge on the robot appearance is used in the method, and it was shown that the obtained low-dimensional space reflected the spatial relation between the robot hand and the object.

2. Integration of multiple sensor spaces with mapping to manifold

In this section, a motion generation method using an integration of multiple sensor spaces is presented. Multiple sensors are often required to realize thorough understanding about the environment, but they often do not overlap with each other; in the case of visual recognition as an example, occlusion and restriction of the viewing range often cause an incomplete state identification. In the case of tactile and proximity sensors, their detection ranges are limited and provide useful information only in limited cases, when they are close to objects or environments. On the other hand, occlusion often occurs when the sensor is close to an object. Robots can identify their surroundings by integrating multimodal sensors, but it causes a

problem of integrating information of multiple different sensors whose detection ranges do not overlap among each other.

A standard way to integrate multiple sensor information in robotics is to rely on sensor models and calibrations, but they require a preparation cost by human designers. For realizing highly adaptive autonomous robots, following properties are required:

- Integration of multimodal sensor spaces, each of which has its specific sensing range (possibly not overlapping with each other).
- Relying only on sensing and actuating information of the robot itself, without using world coordinate system models.

A motion generation framework based on multiple sensors with limited sensing ranges has been presented in [7]. In order to integrate two sensor spaces, an idea of extending a sensor space was proposed, borrowing an idea from diffusion-based learning [8, 9]. The characteristic of the proposed framework is that it can generate desired trajectory and motion without a problem-specific knowledge. It is known that the similar class of problems has been discussed using partially observable Markov decision processes (POMDPs) [10–13]. The proposed framework does not take the noise or perceptual aliasing into account, but the proposed framework is simpler.

2.1. Problem definition of motion generation with multiple sensors

Let $x \in X$ denote the state of the robot system, where $X \subset \mathbb{R}^n$ denotes the state space. Observation vectors are denoted by $s \in s^{(1)}, \dots, s^{(h)} \in s^{(h)}$, where h denotes the number of sensors and $s^{(i)} \subset \mathbb{R}^n$ denotes the observation variable space for sensor i . The control input (motor command) to the system is denoted by $u \in \mathbb{R}^n$. The dynamics of the system is expressed as

$$\dot{x} = F(x)u, F(x) \in \mathbb{R}^{n \times n}, \quad (1)$$

where $F(x)$ is a smooth function, which will be approximated locally. Each sensor's sensing range is limited. $X^{(i)} \subset X$ denotes a subset of state space where sensor i is in its sensing range. Mapping from $X^{(i)}$ to $s^{(i)}$ is assumed to be injective and smooth, where all mappings are unknown. It is also assumed that there is no noise in the observation and the robot can judge whether each sensor is in its sensing range.

The task of the robot is to move from an initial configuration x_{start} to x_{goal} , where the information of the target configuration is given as an observation vector sensed at x_{goal} . That is, the target sensor value is given to the robot as $s_{\text{goal}}^{(h)}$ where j satisfies $x_{\text{goal}} \in X^{(j)}$. As indicated in **Figure 1**, a single sensor does not cover both x_{start} and x_{goal} , nor is it guaranteed that the sensing range of one sensor does not overlap with that of another sensor. Thus, the robot must find a trajectory that goes through a subset of the state space where no sensor signal can be observed.

The second aspect of the problem is that observation variable itself contains redundancy. Let m denote the dimension of the observation variable and the redundancy means $m > n$, which

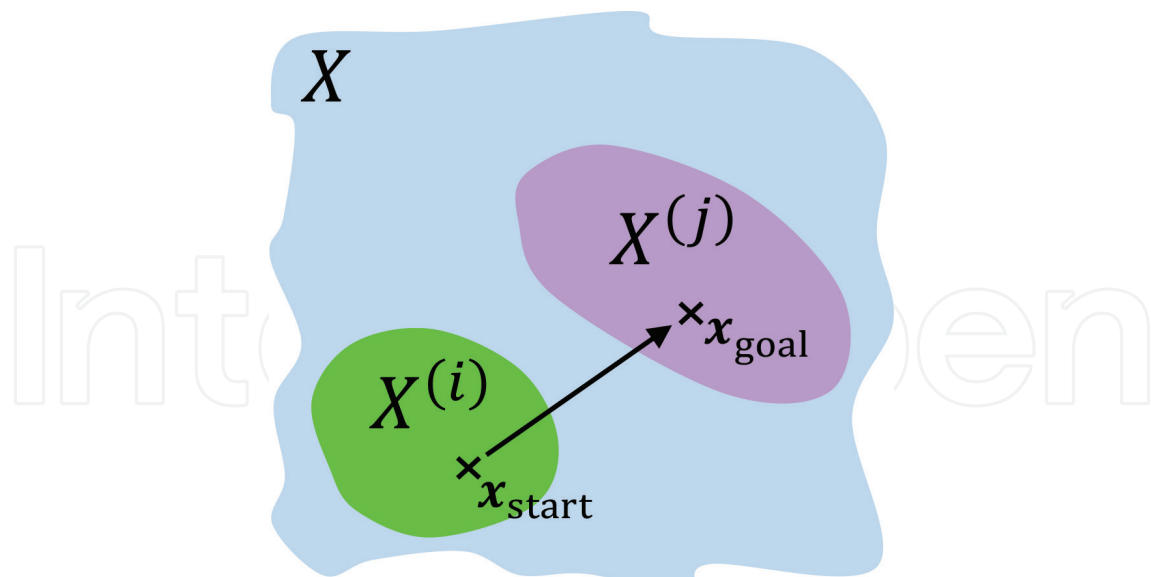


Figure 1. Motion from a sensing range of a sensor to another range.

is depicted in **Figure 2** with the case $m = 3$ and $n = 2$. The observation variables are constrained on a two-dimensional manifold. Knowing the fact of constraint on a manifold, lower dimension can be obtained by approximating the manifold.

2.2. Integration of multiple sensor spaces

The robot first acquires the mapping from the control input to an observation vector by collecting samples by random motion within each sensor's detection range. The basic idea of the integration is to first extend the mapping from outside the sensing range as indicated in **Figure 3**. The robot repeats motion in and out of the sensing range and compares the resultant observation and a predicted observation. That is, the robot estimates the observation using the information of its sequential motion and the input-observation mapping.

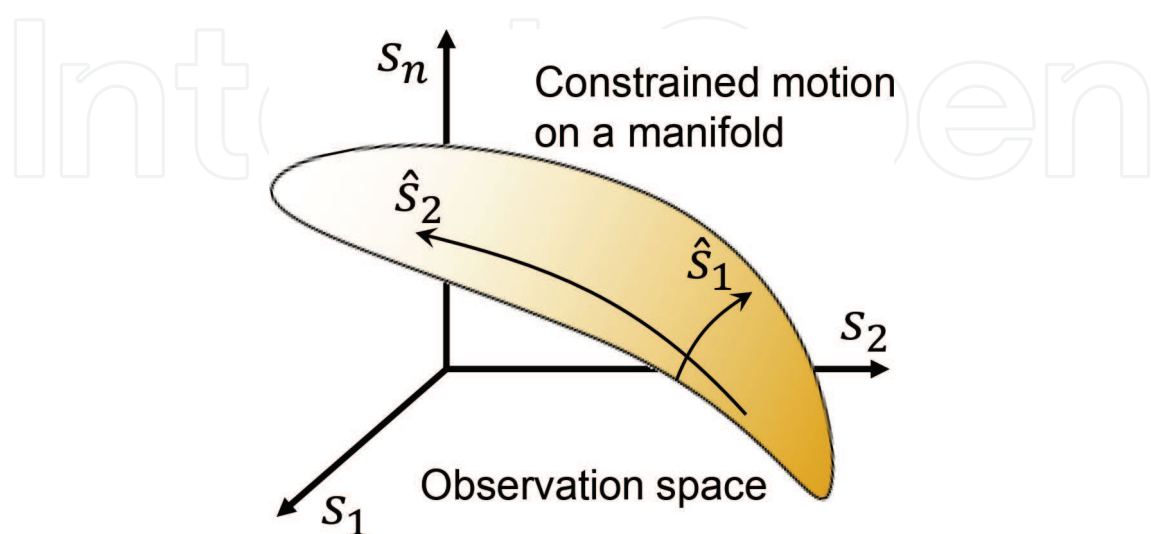


Figure 2. Redundant sensor information and constraint on a manifold.

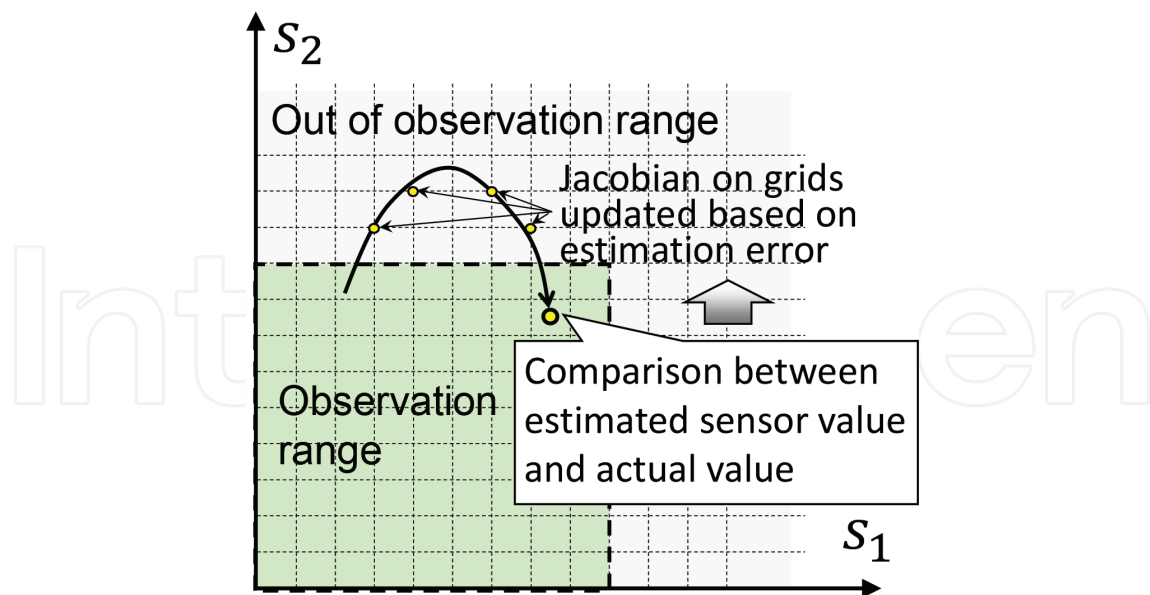


Figure 3. Estimation of Jacobian of observation variable dynamics outside observation range of a sensor.

The process of extension of a sensor space can be understood as a construction of a “virtual” observation space. The virtual observation space overlaps with another sensor space. When a task to reach a destination is given to the robot, it generates a motion from the current sensor space to the other sensor space including the destination, which is based on the representation of the virtual observation space. This framework basically works on the basis that the dimensions of the sensor spaces are equal. To relax the condition, we also discuss the way to deal with a case of redundant sensor space where an observation vector has higher dimension than the state vector.

2.3. Dimension reduction of the observation vector

Dimension reduction of sensor variable is based on a grid-based parameterization, as shown in **Figure 4**. Basic idea of the parameterization is similar to an active contour model used in image processing [14]. The nodes in a two-dimensional grid fit along the surface of samples by minimizing an energy representing closeness to the samples. By extending and fixing the nodes on the ends of the grid to the end of samples, iterative updates minimizing the energy lead the grids to fit the samples while spreading to cover the sample region. Once the lower-dimensional grid is created, it is used to parameterize the original sensor signal by another vector, in the example case, in two-dimensional vector.

2.4. Motion generation by integrating two observation spaces

Using the extrapolation of Jacobian in the observable region to outside the viewing range, virtual observation variables can be obtained. As shown in **Figure 5**, the robot starts motion from a viewing range of sensor i . The target is given as a variable of sensor j . Using the extrapolation, the closest grid in the virtual sensor space of sensor i can be calculated. First, the robot is controlled to aim at the grid in the virtual sensor space using Jacobian of sensor i . After it

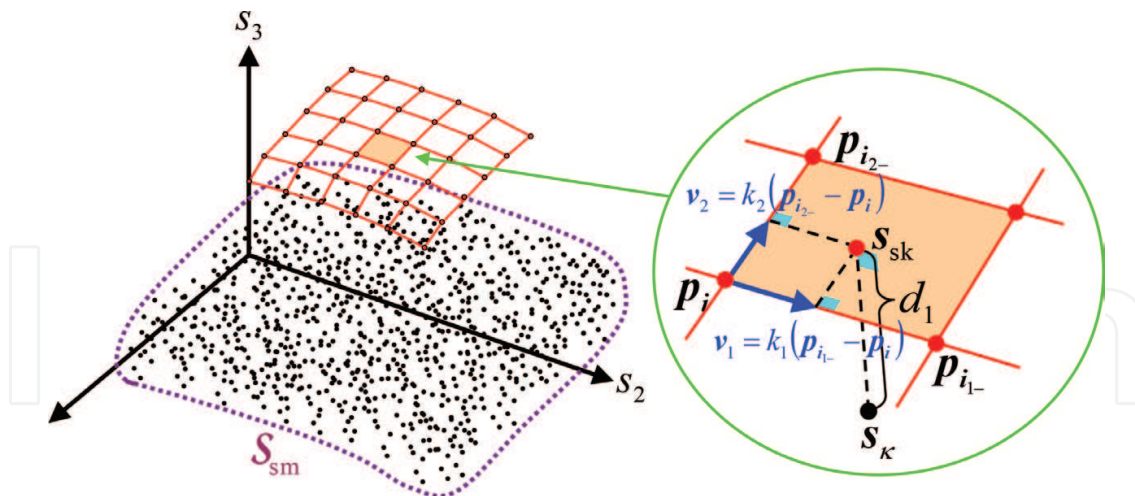


Figure 4. An example of a grid-based parameterization of sensor space with three-dimensional sensor variable. Input in the three-dimensional space is mapped onto a two-dimensional vector.

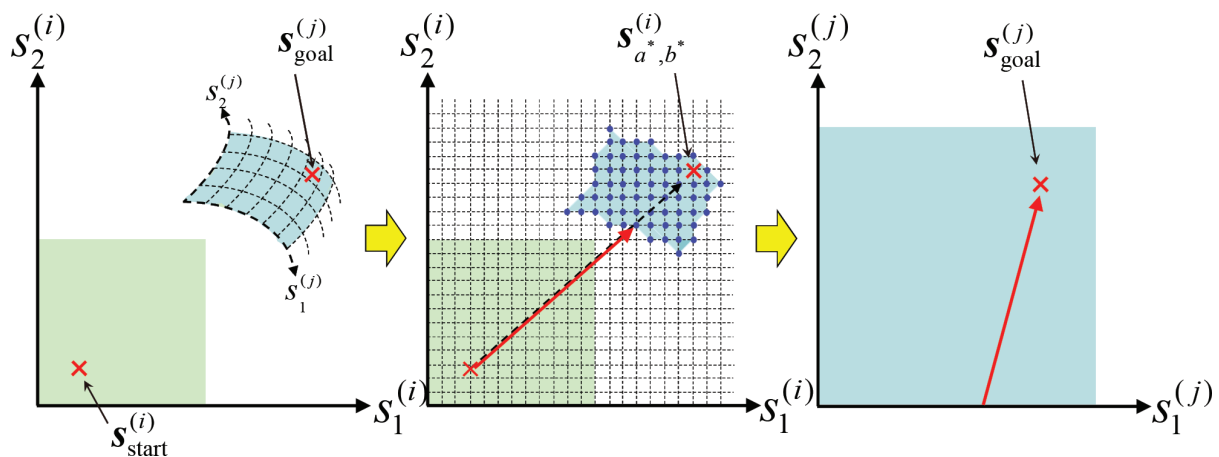


Figure 5. Motion generation by integrating two observation space, starting from a sensor space aiming at another sensor space and finally reaching the target.

reaches the viewing range of sensor j , it switches to the Jacobian of sensor j to finally reach the precise target position in the space of sensor j .

2.5. Simulation results

Consider a mobile robot with five proximity sensors, as shown in **Figure 6**. The robot is equipped with three proximity sensors on the front of its body that are grouped as sensor 1. The robot has two proximity sensors on its right side. They are grouped as sensor 2. A wall that has an infinite length is located in the environment. Each proximity sensor provides a value that is proportional to the distance to the wall. When the distance is longer than its sensing range, the wall cannot be detected. This situation is assumed to be detected by the robot. The control input to the system is the angular velocities of the two wheels.

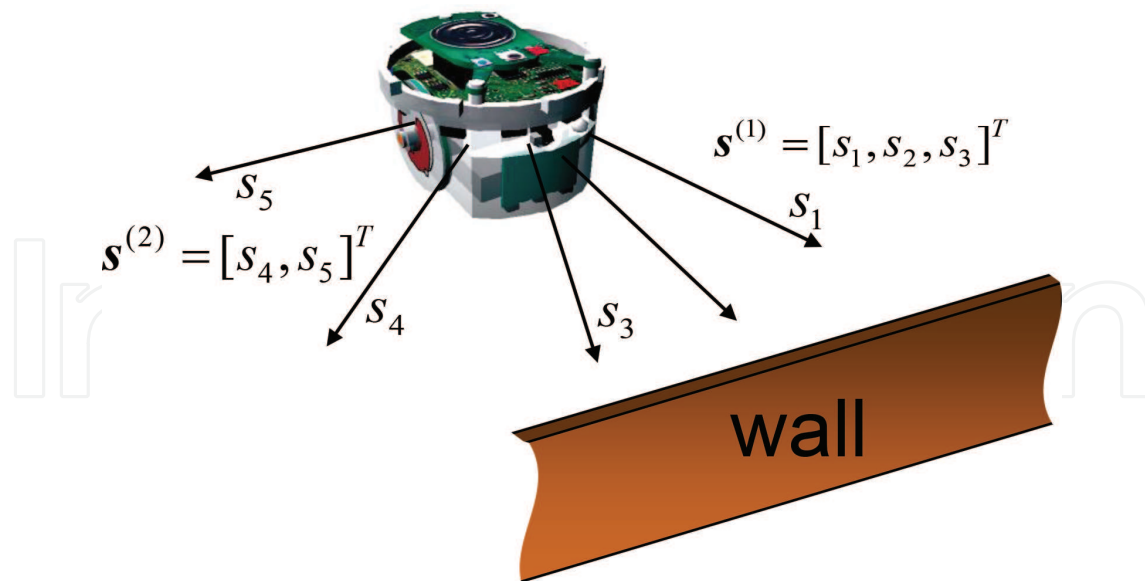


Figure 6. Mobile robot navigation problem where five distance sensors are available to detect a relative configuration against a wall.

In this problem setting, the relative position of the robot to the wall can be expressed by two parameters; its distance to the wall and its relative orientation to the wall. When the robot is moving parallel to the wall, sensor 2 detects the configuration of the robot. In this case, sensor 1 is out of its sensing range. Sensor 1 provides configuration information when the robot is facing the wall. However, in this case, the observation variables $\mathbf{s}^{(1)} = [s_1, s_2, s_3]^T$ are redundant for the purpose of specifying the 2-DOF configuration relative to the wall. The mapping method of manifold is applied to the space of sensor 1. The radius of the wheels is 0.02 (m), and the distance between the two wheels is 0.04 (m). The initial state of the robot is set where the robot faces perpendicular to the wall at a distance of 1 (m), and final destination of the robot is specified so that the robot comes close to the wall where only sensor 2 is in its detection range.

Figure 7(a) shows samples of the observation variables of sensor 1, which was obtained by the offline random data collection. The three-dimensional vectors are distributing on a

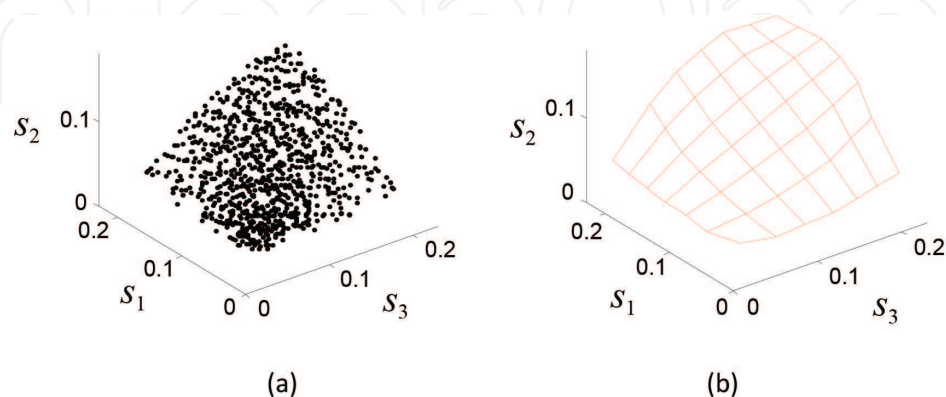


Figure 7. Collected sensor variables with three distance sensors (a) and estimated two-dimensional manifold based on the samples (b).

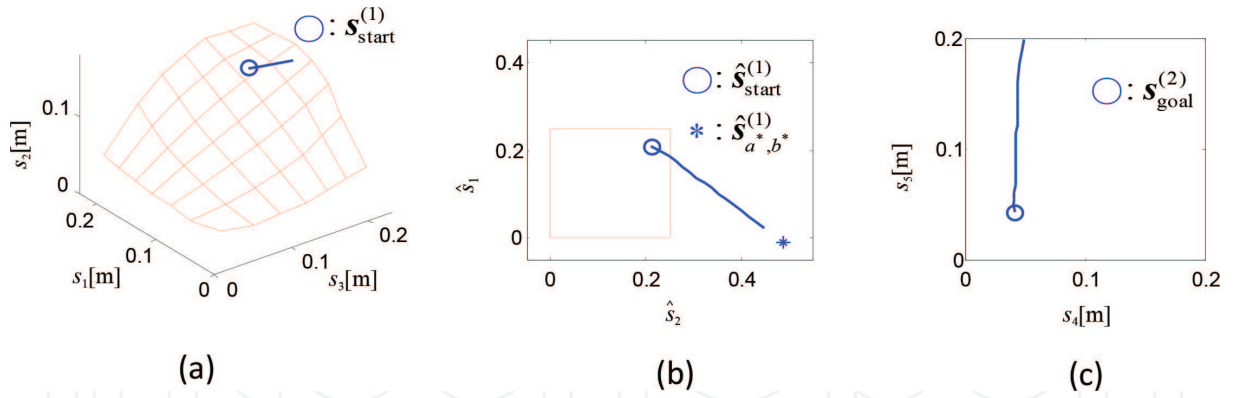


Figure 8. Realized trajectories of the robot in sensor space 1 (a), virtual sensor space of sensor 1 (b), and the viewing range of sensor 2 (c).

two-dimensional surface of the redundancy described above. Nodes of the approximation surface were initially located around the center of the samples. The approximation surface obtained by the proposed method is depicted in **Figure 7(b)**, where the nodes correspond to crosses on the curves. It can be confirmed that the nodes covered the samples by spreading and fitting them.

The trajectory obtained by the proposed method in the observation variable spaces is shown in **Figure 8**. The line drawn on the approximation surface in **Figure 8(a)** indicates the initial part of the trajectory in the observation variable space for sensor 1, where a circle in the figure indicates the initial configuration. **Figure 8(b)** shows the trajectory drawn in the space of sensor 1 obtained by the proposed dimension-reduction method. **Figure 8(c)** shows the last part of the trajectory in the space of sensor 2.

The trajectory generated in the world coordinate by the robot is depicted in **Figure 9**. The line of $y = 0$ indicates the wall. The initial configuration of the robot is apart from the wall and it

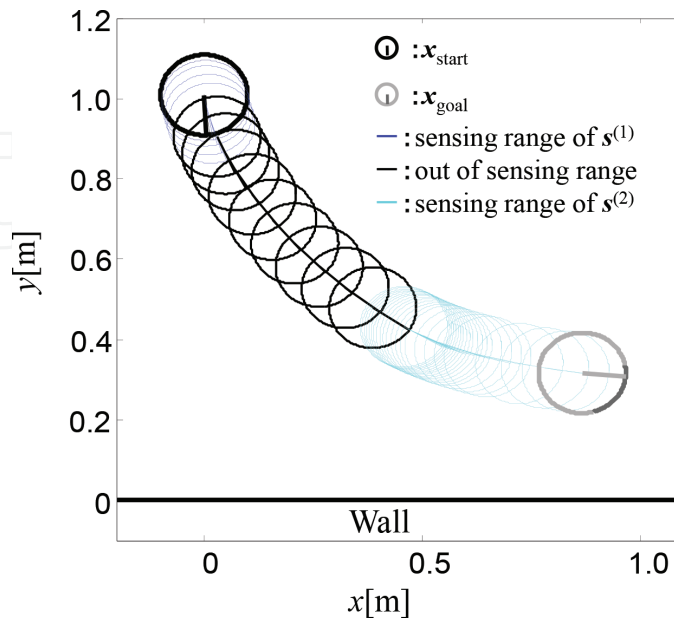


Figure 9. Trajectory of the mobile robot in the world coordinate view.

finally reached the target configuration, directing parallel to the wall. It can be seen that there is an intermediate part in the trajectory, where none of the distance sensors were detecting the wall, as drawn in thick lines in the trajectory.

It was assumed in the proposed method that the robot knows the dimension of the state vector. This problem can be resolved by applying statistical methods such as principal component analysis (PCA) [15], which allows to detect the appropriate dimension of the robot system's dynamics. Though PCA is a linear framework, which is valid only in the case where linear dimension reduction can be applied to the whole state space, nonlinear extensions of the dimension-reduction methods have been also developed, such as Kernel PCA [16] and ISOMAP [17]. The surface-approximation scheme applied in this chapter for the dimension-reduction problem can be replaced to other nonlinear mapping methods, which will be one of our future works.

3. Manifold learning approach toward constructing state representation for robot motion generation

Monocular and stereo cameras are widely used as external sensors for robot systems. In the real-world application of robot systems, however, measurement of 3D configurations of objects suffers from the following difficulties:

1. 3D configuration measurement, in general, inherently requires precise measurement of the shape of an object, but the whole shape of an object cannot be measured directly because the process is normally unilateral.
2. It is very important for object manipulation of a robot that the spatial relation between a robot and an object is precisely identified. But while the robot hand is approaching to the object and getting close to it, occlusion is very likely to occur.
3. In real applications, objects very often deform by contact with the robot, which requires specific model for mathematical analysis. But it is difficult to precisely model the deformation.

In the research field of developmental robotics, measurement of the 3D configuration in the world coordinate is not regarded as a sole way to represent the state for a robot. If a robot can build a suitable representation of its environment based by its own way, the total process of robot recognition and motion generation will be freed from the problems mentioned above (e.g., see [18] as a learning approach).

Thus, an approach to the interest of building a representation of a robot from images for motion planning and control in an adaptive way without any predefined knowledge [19] is presented in this section. To consider relation between the robot and its environment, image features based on scale invariant feature transform (SIFT) [20] are used. As a related research, an image feature-based learning of robot behavior was presented [21]. However, it did not deal with relation between an object and the robot with a quantitative representation. In the presented method, a manifold learning method is applied to acquisition of state

representation. It allows not only to classify state of the robot but also to evaluate closeness to a certain situation. In addition, it is verified that the representation acquired by the method is used for motion generation of collision avoidance.

As a means of manifold learning, locally linear embedding (LLE) [6] is used. The manifold learning is suitable because the system dynamics property can hold only in a local region in the problem of robot motion generation. A vector generation based on SIFT features matching is proposed for the application of LLE to deal with the problem that keypoints of SIFT are not consistently observed throughout the image sequences. The proposed method is evaluated using a humanoid robot with real images after verification of LLE state representation generation with simulated images.

3.1. Problem definition of manifold learning from an image

Figure 10 shows images obtained by CCD camera attached at the head of a robot. These images are input to the system. Humanoid robot NAO [22] is considered in the experiment. The images contain part of the body (arm) of the robot, an object that has possibility to contact with the robot, and other objects that are not affected by the robot motion (background). Shoulder roll joint and shoulder pitch joint are controlled, while other two joints are fixed throughout the experiment. This implies that the motion of the robot arm is constrained on a plane that is vertical to optical axis of the CCD camera.

The right hand of **Figure 10** also shows image features extracted from the images as depicted by circles. Keypoints of SIFT are used as image features. No explicit knowledge on properties of image features is assumed in the problem. That is, the robot does not have label information of the object, backgrounds, or robot's body in the image in advance. The robot collects images while moving its arm randomly. Position of the object is also differed irrelevantly to the configuration of the robot arm.

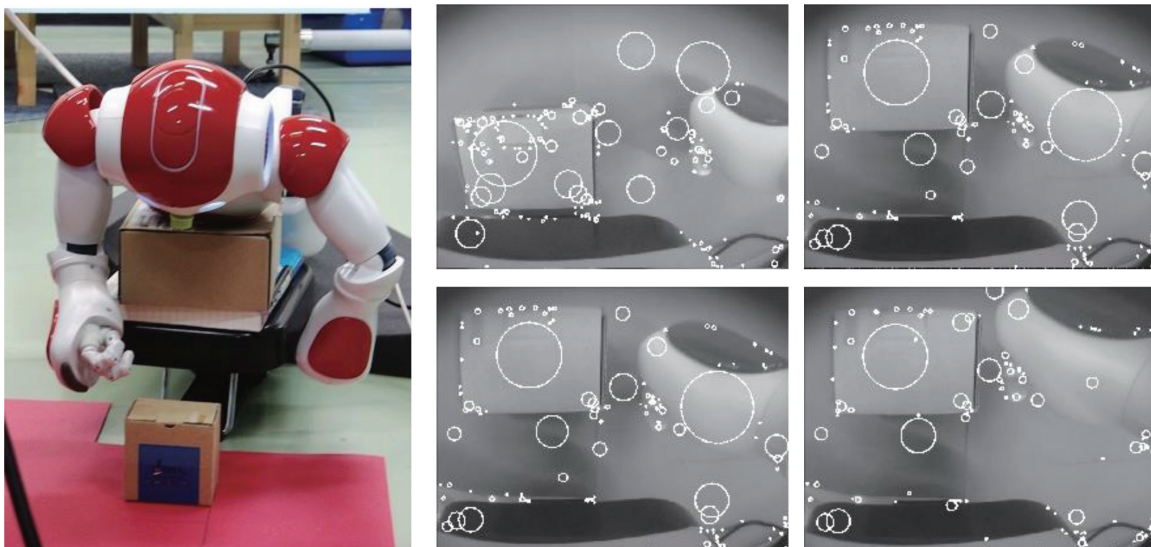


Figure 10. Experimental setup of a humanoid robot with an object (left). Images and SIFT keypoints extracted as circles with different scales indicated by circles (right).

The objective for the robot is to construct a space that provides the following utilities:

1. Estimation of closeness between its hand and the object
2. Prediction of collision between its hand and the object

The first utility allows the robot to plan its motion so that its hand does not come too close to the object while the robot tries to reach some configuration, avoiding collision with obstacles. The second utility is expected to contribute to the ability to predict collision prior to its motion by integrating it into other techniques, for example, prediction of robot's hand in the image space.

3.2. Manifold learning based on SIFT image features

Manifold learning by LLE is applied to the vectors represented by positions of SIFT keypoints. Each keypoint contains 128-dimensional feature vector that is used to classification and matching to the keypoints in other image frames. By the matching process, a keypoint can be tracked through multiple image frames given that it is extracted in those images. However, in the application of robot motion sequence, each feature vector corresponding to a keypoint is not consistent through sequences of image frames. The arm, which consists of serial links, inevitably changes its posture while it is moving toward a certain configuration. By assuming that each keypoint tracks a certain part of the arm, we proposed a method for matching and labeling using self-organizing map (SOM) [23].

Although feature vectors of a keypoint differ by the change of the robot's configuration in the image frames, it is likely that those feature vectors in images with small differences in image pixel level are similar. By using topological neighbor of SOM generated by image pixel information, correspondence between keypoint labels can be found. By finding correspondence between neighbor nodes, labels that correspond to the same part of the real world are integrated into one label.

3.3. Motion generation based on manifold learning

Dynamic programming with discrete state representation [10] is applied for motion generation. The state for motion generation is defined by the joint angle space. The discrete state is given by discretizing the joint angles of the robot two-dimensional grids. Actions are defined as four directional transitions from a grid to its adjacent grids. Reward is defined as 0 for reaching the desired configuration, -100 for colliding with the object, and -1 at every step otherwise. Collision with the object is predicted using the obtained LLE representation as described below.

3.4. Simulation results

We first tested basic property of LLE in conditions similar to the experimental problem setting. Virtual keypoints were generated as shown in **Figure 11(a)**. As an assumption, an object and the robot hand is captured in an image frame with the size of 400×400 [pix]. There were 10 keypoints to be detected on the object, 10 on the robot hand, and 5 in the background. Both the positions of the object and the hand were varied randomly with uniform distribution.

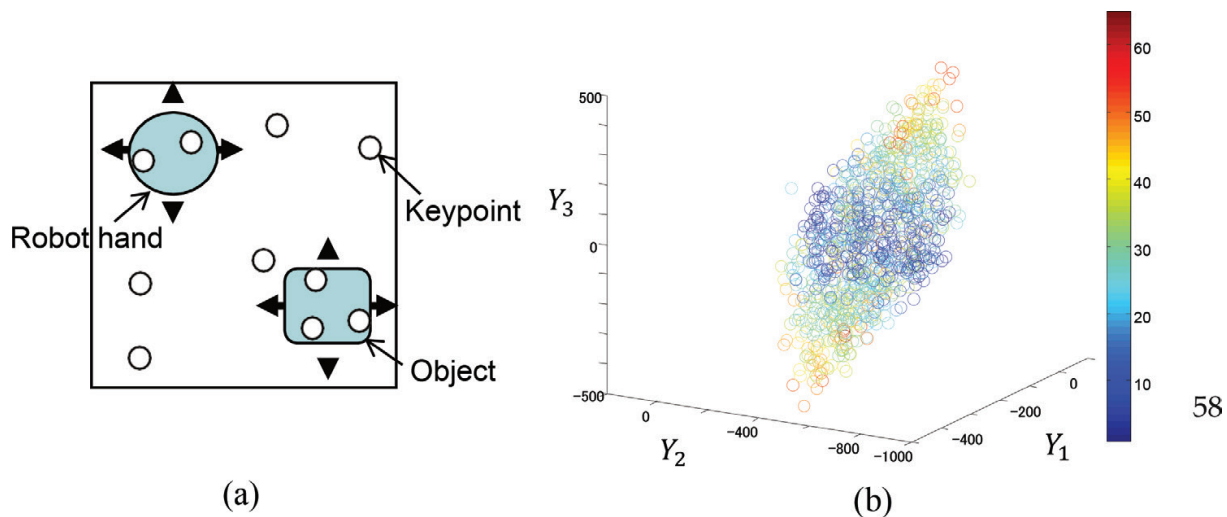


Figure 11. Simulated keypoints (a) and result of LLE with distance information between robot hand and object (b).

Total number of images was 1000. Number of keypoints was 25. To simulate matching error of keypoints, position information of 10% of the keypoints in the data vector was removed.

The result of mapping by LLE is depicted in **Figure 11(b)**. Y_1 , Y_2 and Y_3 in the figure correspond to low-dimensional vector y and hence they do not have units. The colors of the points denote distances between the object and the hand in the corresponding images, where the original distance information in pixel with maximum 550 pixel was converted to 64 levels. It can be seen in the figure that one direction in the feature space reflects the distance between the object and the hand.

3.5. Experiment of LLE mapping and motion generation with real images

The three-dimensional mapping constructed by the proposed method is depicted in **Figure 12**. Each point, indicated by a circle or a cross, indicates a vector obtained by converting the image feature vector by LLE. A cross denotes an image corresponding to a situation where the hand contacts with the object. A circle denotes an image without any contact. It can be seen that in the space, crosses are concentrating around a certain region. Distance between the object and the hand, however, could not be clearly seen in the obtained map.

For verification, some test images that are independent from the training process of LLE mapping generation were mapped onto the generated space. Test samples are drawn by boxes in the figure. Corresponding images are also displayed. It is observed that the image with its robot hand, the most distant from the object, is located in the space at the furthest position from the region of the dense crosses. Images with its hand closer to the object are located also closer to the “contact” region. But there is a jump at the last step to contact with the object into the region with dense crosses. Thus, the spatial relation between the hand and the object was reflected to a certain level, but not directly reflecting the distance between the hand and the object in the real world.

Classification of collisions was also evaluated based on the generated map information. Using the mapping collision between the hand and the object was predicted by whether an image is included in the sphere whose center is the average of the samples indicated by the crosses. The optimal radius was set as $r = 0.74$, which was found empirically so that the discrimination

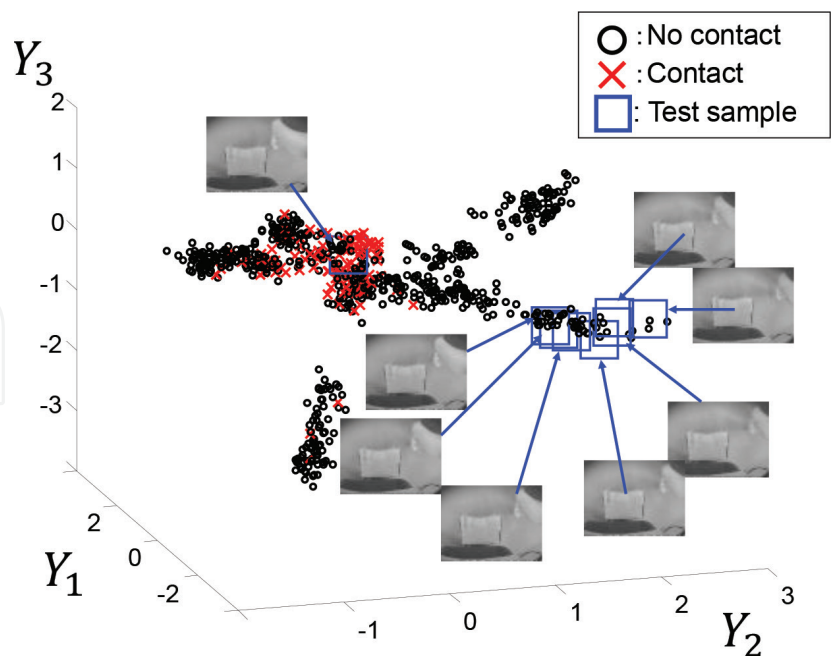


Figure 12. LLE mapping by real images with test samples.

performance was the best. The classification result is shown in **Table 1**. For comparison between linear and nonlinear methods, a linear mapping was also implemented. The classification result based on the mapping principal component analysis (PCA) is depicted in **Table 2**. It can be seen that nonlinear mapping brought conspicuous difference of classification performance.

A sequence of snapshots of motion generated by DP is shown in **Figure 13**. Grid sizes for the discrete state space were set as 8×12 . Collision was predicted by a correct recognition result for images adopted in **Table 1**. (1) in the figure denotes the initial configuration of the robot hand. The tip of the hand is located above the object in (11), corresponding to the target configuration. It can be seen that the robot hand could reach a destination while avoiding collision with the object, given that an appropriate evaluation of closeness (or collision) to the object is achieved.

	Collision (%)	No collision (%)
Recognized as collision	95/115 (82.6)	111/617 (18.0)
Recognized as no collision	20/115 (17.4)	506/617 (82.0)

Table 1. Prediction of collision with LLE.

	Collision (%)	No collision (%)
Recognized as collision	63/115 (54.8)	132/617 (21.4)
Recognized as no collision	52/115 (45.2)	485/617 (78.6)

Table 2. Prediction of collision with PCA.

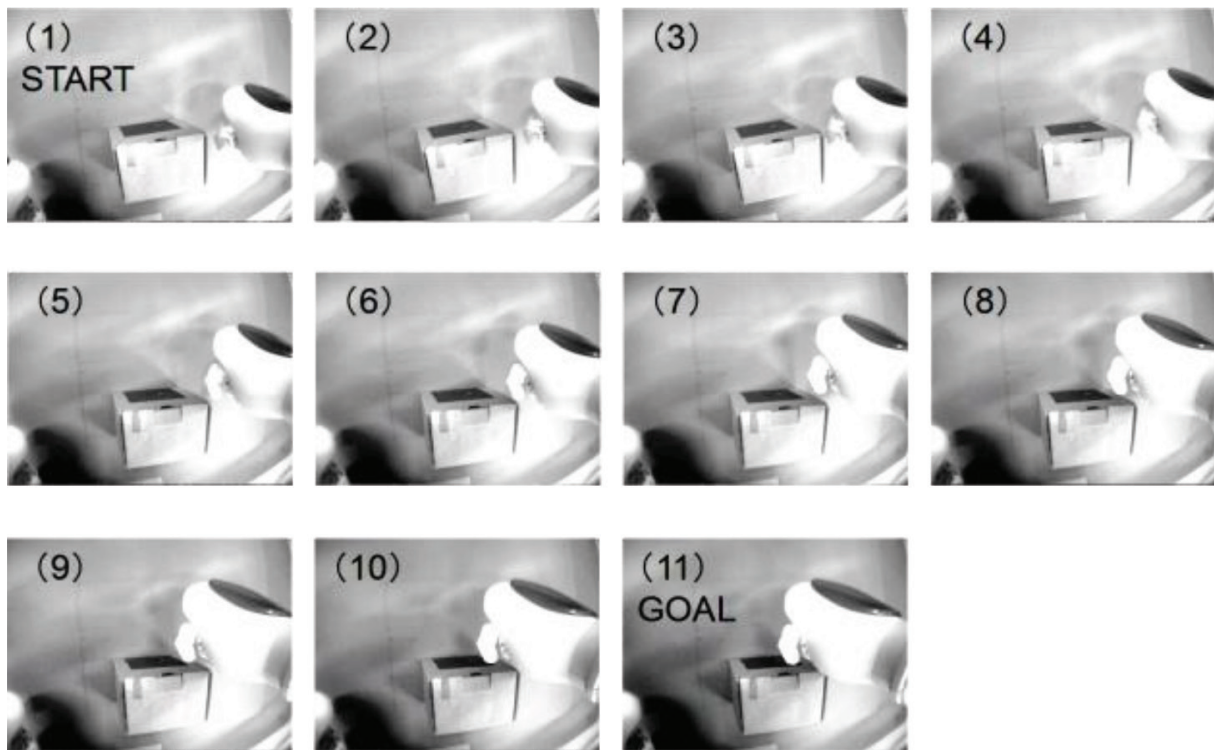


Figure 13. Snapshots of motion a motion sequence achieved by the proposed motion generation.

4. Conclusion

Two kinds of application of manifold were presented in this chapter. In the first application, coordinate systems obtained from sensor signals are directly used for motion control of the robot. In the second application, an intermediate representation, spatial relation between the robot hand and the object, was built using a manifold learning method. One important advantage of these approaches, in comparison with the end-to-end motion learning approaches such as deep learning (e.g., [24]), is that we can analyze and evaluate the obtained representation. In order to apply the approach of manifold learning to more complex robot motion problems, it will be required to consider multiple resolutions, disappearance of features (as discussed in [25]), multiple relations among variables (e.g., discussed in [26]), and connecting different modalities with discontinuous dynamics (such as contact and noncontact switching).

Author details

Yuichi Kobayashi

Address all correspondence to: kobayashi.yuichi@shizuoka.ac.jp

Shizuoka University, Hamamatsu, Japan

References

- [1] Asada M, MacDorman K, Ishiguro H, Kuniyoshi Y. Cognitive developmental robotics as a new paradigm for the design of humanoid robots. *Robotics and Autonomous Systems*. 2001;**37**:185-193
- [2] Stoytchev A. Some basic principles of developmental robotics. *IEEE Transactions on Autonomous Mental Development*. 2009;**1**(2):122-130
- [3] Thrun S, Burgard W, Fox D. *Probabilistic Robotics*. 2005
- [4] Fox D, Thrun S, Burgard W, Dellaert F. Particle filters for mobile robot localization. In: Doucet A, de Freitas N, Gordon N, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York, NY: Springer Verlag; 2001. pp. 499-516
- [5] Elfes A. Using occupancy grids for mobile robot perception and navigation. *Computer*. 1989;**22**(6):46-57
- [6] Saul LK, Roweis ST. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*. 2003;**4**:119-155
- [7] Kobayashi Y, Kurita E, Gouko M. Integration of multiple sensor spaces with limited sensing range and redundancy. *International Journal of Robotics and Automation*. 2013; **28**(1):31-41
- [8] Luo ZW, Ito M. Diffusion-based learning theory for organizing visuo-motor coordination. *Biological Cybernetics*. 1998;**79**:279-289
- [9] Luo ZW, Ando H, Hosoe S, Watanabe K, Kato A. Spatial generalization of optimal control for robot manipulators. *Journal of Robotics and Mechatronics*. 2001;**13**(5):533-539
- [10] Sutton R, Barto A. *Reinforcement Learning*. Cambridge, MA: MIT Press; 1998
- [11] Kaelbling LP, Littman ML, Cassandra AR. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*. 1998;**101**:99-134
- [12] Andrew McCallum R. Instance-based utile distinctions for reinforcement learning with hidden state. In: *Proceedings of the 12th International Machine Learning Conference*. 1995
- [13] Wierstra D, Foerster A, Peters J, Schmidhuber J. Solving deep memory POMDPs with recurrent policy gradients. In: *Proceedings of the International Conference on Artificial Neural Networks ICANN'07*. 2007
- [14] Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models. *International Journal of Computer Vision*. January 1988;**1**(4):321-333
- [15] Fukunaga K. *Introduction to Statistical Pattern Recognition*. San Diego, CA, USA: Academic Press Professional, Inc.; 2013

- [16] Schölkopf B, Smola A, Müller K. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*. 1998;**10**(5):1299-1319
- [17] Tenenbaum JB, de Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. *Science*. 2000;**290**(5500):2319-2323
- [18] Prankl J, Zillich M, Vincze M. 3d piecewise planar object model for robotics manipulation. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. 2011. pp. 1784-1790
- [19] Kobayashi Y, Matsui R. Manifold learning approach toward constructing state representation for robot motion generation. *Transactions on Computational Collective Intelligence XXIV, LNCS*. 2016;**9770**:101-116
- [20] Lowe DG. Object recognition from local scale-invariant features. In: *Proceedings of IEEE International Conference on Computer Vision*. Vol. 2. 1999. pp. 1150-1157
- [21] Kobayashi Y, Okamoto T, Onishi M. Generation of obstacle avoidance based on image features and embodiment. *International Journal of Robotics and Automation*. 2012;**24**(4): 364-376
- [22] <https://www.softbankrobotics.com/emea/>
- [23] Kohonen T. *Self-Organizing Maps*. Tokyo, Japan: Springer Press; 1995
- [24] Mnih V et al. Human-level control through deep reinforcement learning. *Nature*. 2015;**518**:529-533
- [25] Somei T, Kobayashi Y, Shimizu A, Kaneko T. Clustering of image features based on contact and occlusion among robot body and objects. In: *Proceedings of the 2013 IEEE Workshop on Robot Vision (WoRV2013)*. 2013. pp. 203-208
- [26] Kobayashi Y, Hosoe S. Planning-space shift motion generation: Variable-space motion planning toward flexible extension of body schema. *Journal of Intelligent and Robotic Systems*. 2011;**62**(3):467-500