

2002

Skill acquisition in transfer of manipulation skills from human to machine through a haptic virtual environment

Y. Chen

University of Wollongong

F. Naghdy

University of Wollongong, fazel@uow.edu.au

Publication Details

This article was originally published as: Chen, Y & Naghdy, F, Skill acquisition in transfer of manipulation skills from human to machine through a haptic virtual environment, IEEE International Conference on Industrial Technology (ICIT '02), 11-14 December 2002, 1, 337-342. Copyright IEEE 2002.

Skill acquisition in transfer of manipulation skills from human to machine through a haptic virtual environment

Abstract

A new paradigm for programming a robotics manipulator is developed. It is intended that the teaching of the machine will begin with the necessary skills being demonstrated by the human operator in a virtual environment with tactile sensing (haptics). Position and contact force and torque data generated in the virtual environment combined with a priori knowledge about the task is used to identify and learn the skills in the newly demonstrated tasks and then to reproduce them in the robotics system. The peg-in-hole insertion problem is used as a case study. The overall concept is described. The methodologies developed to build the virtual environment and to learn the basic skills are explained. The results obtained so far are presented.

Keywords

haptics, skill learning, virtual reality, peg-in-hole insertion

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Chen, Y & Naghdy, F, Skill acquisition in transfer of manipulation skills from human to machine through a haptic virtual environment, IEEE International Conference on Industrial Technology (ICIT '02), 11-14 December 2002, 1, 337-342. Copyright IEEE 2002.

Skill Acquisition in Transfer of Manipulation Skills from Human to Machine through a Haptic Virtual Environment

Y. Chen and F. Naghdy*

School of Electrical, Computer and Telecommunication Engineering,
University of Wollongong, NSW, 2522, Australia. Fax : +61 42 21 3236

Email: yc12@uow.edu.au

* Phone : +61 42 21 3398, Email: f.naghdy@uow.edu.au

Abstract

A new paradigm for programming a robotics manipulator is developed. It is intended that the teaching of the machine will begin with the necessary skills being demonstrated by the human operator in a virtual environment with tactile sensing (haptics). Position and contact force and torque data generated in the virtual environment combined with a priori knowledge about the task is used to identify and learn the skills in the newly demonstrated tasks and then to reproduce them in the robotics system. The peg-in-hole insertion problem is used as a case study. The overall concept is described. The methodologies developed to build the virtual environment and to learn the basic skills are explained. The results obtained so far are presented.

Keywords: Haptics, skill learning, virtual reality, peg-in-hole insertion

1. Introduction

Robotics manipulators have been primarily employed to perform a particular task through programming. The programming methods developed so far can be grouped into four categories of text programming, off-line simulation-based programming, inductive learning and teaching by guiding.

Text programming can be applied to complex applications, but the development time is long and special skills and much effort are required to produce a complete program. This has resulted in the development of task level robot languages [1][2]. But generic task level languages have proved to be quite intensive in code and computing time.

Off-line simulation-based methods usually integrate text-programming and model-based motion planners in one common platform [3][4]. The approach is powerful but requires special hardware and a complete description of the real world, both of which are costly.

In inductive learning, a robot arm masters appropriate motion and sensing strategies through trial and error [5]. This is an effective method when it is used to refine other programming methods.

In "teaching by guiding" a human operator drives the robotics arm in the real world to perform the task while the characteristics of the motion are recorded. In spite of its simplicity, the method is not generic, flexible or robust, and is not applicable to complex tasks. It cannot accommodate extensive sensory interaction and can be dangerous for the operator.

There have been a number of attempts to overcome some of the shortcomings of "teaching by guiding" approach. Summers and Grossman [6] embedded the collection of the sensory information and interaction with the operator in the task instruction procedure. Asada and Assari [7] extracted the control rules to perform a particular assembly motion from the position and force data generated during operation of a human operator. Sator and Hirai [8] integrated direct teaching with task level languages through master-slave manipulators.

The concept of "teaching by showing" has been another extension of "teaching by guiding", in which a robotics system learns a particular task by watching a human operator performing it. The learning methodologies were initially developed for computer scene understanding [9], and automatic perception of actions [10][11].

Some recent developments have significantly advanced the "teaching by showing" approach in programming a robotics manipulator. Ikeuchi and Sehiro developed a system that could extract fine motion sequence from transitions of face contact states obtained by a range sensor [12]. Haas integrated a symbolic recogniser and play back module using a visual servo for two-dimensional pick and place operations [13]. Yamada and Uchiyama conducted a study to determine essential features of human physical skills based on multi-sensory data and the possibility of transferring them to robots by focusing on two tasks of crank rotation and side matching [14].

Kuniyoshi et al developed a robotics system that could learn reusable task plans in real time by watching a human performing assembly tasks [15]. The method was based on visual recognition and analysis of human action sequences. The effectiveness of the method was demonstrated for a block assembly task.

Direct transfer of skills from a human operator to a machine in an interactive environment has been the next stage in the programming and training of a robotics system. In the field of mobile robots, Pomerleau used a 3 layer perceptron network to control the CMU ALVINN autonomous vehicle [16]. Grudic and Lawrence used an approximation method as a means for creating the robot's mapping from sensor inputs to actuator outputs in transfer of skills to a mobile robot [17].

In acquisition of manipulation skills, particularly in constrained motion, the work carried out by Kaiser and Dillman [18] is of significance. The work proposes a general approach to the acquisition of sensor-based robot skills from human demonstration. An adaptation method

is also proposed to optimise the operation with respect to the manipulator. The method is validated for two manipulation tasks of the peg-in-hole insertion and opening a door.

Handleman and Lane [19] have carried out some preliminary work on a knowledge-based “tell” approach to describe the task to be carried out by the robot and the corrective control measures to be taken up. The task is defined by a rule-based goal directed strategy. The proposed method has been verified through computer simulation only for a typical peg-in-hole insertion problem. The development of the rule-based system has been intuitive and rather complicated. The developed rules are very much context based and have to be built from scratch for any new application.

A new paradigm for programming of robotics manipulator to perform complex constrained motion tasks is being studied. The teaching of the manipulation skills to the machine starts by demonstrating those skills in a haptic-rendered virtual environment. Position and contact force and torque data generated in the virtual environment combined with a priori knowledge about

the task is used to identify and learn the skills in the newly demonstrated tasks and then to reproduce them in the robotics system. The use of the virtual environment will simplify the process, as the training data will be directly extracted from the haptic system [20]. This is also a novel approach to machine training enabling this research to take advantage of recent developments in virtual reality and computer simulation.

The developed approach is studied in the context of the peg-in-hole insertion that represents a large number of assembly tasks. However, the algorithms developed for the system are as generic as possible. They identify the basic manipulation skills performed in an automatic assembly and translate them to trajectories and task schedules for a particular application and robotics manipulator.

The overall approach pursued in this work is presented in Fig. 1. As illustrated in this diagram, the robotics manipulator mimics the behaviour of the human operator by acquiring the skills and producing the machine control action $u^m(t)$ from $y^h(t)$ as illustrated in Fig. 1.

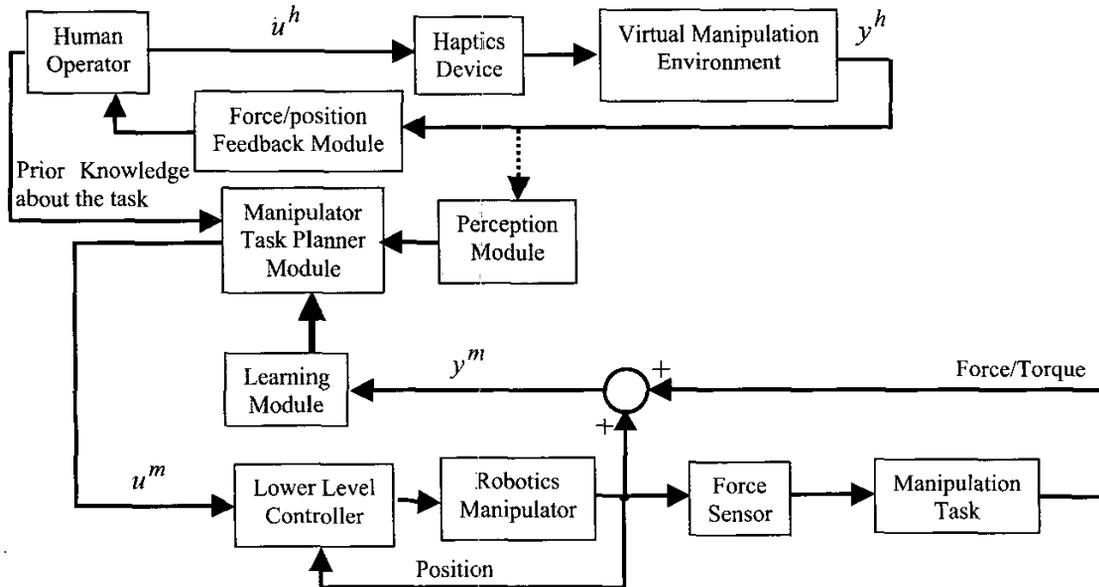


Fig.1 Overall model of the system

2. Virtual Manipulation Environment

The data used by the machine to acquire basic manipulation skills is generated through a haptic-rendered virtual environment. This approach offers a number of advantages compared to other methods of obtaining training data including:

1. The training data (e.g. velocities, angles, positions, forces and torque) can be extracted and recorded directly. This simplifies the data collection process [20].
2. The environment can be easily modified and changed as the manipulation process and its requirements are changed.
3. The risk of breakdown and breakage of the system is very low.

4. Dangerous and costly environments can be easily constructed and simulated.
5. A user-friendly environment for the human operator can be developed.

The concepts and methodologies developed in this work are demonstrated for the peg-in-hole insertion which represents a typical manipulation task in assembly.

The haptic rendering is provided through a 3 degree of freedom generic device called Phantom manufactured by Sensable. It allows users to directly interact with digital objects as they do in the real world. GHOST® SDK, the software operating with Phantom, can handle complex computations and allow developers to deal with simple, high-level objects and physical

properties like location, mass, friction and stiffness [21].

The developed virtual environment is shown in Fig. 2. The peg is coupled with the phantom (i.e. the manipulation point) through a spring-damper system. The peg is a dynamic rigid object in the virtual environment. The force and torque reacted to the peg are transferred to PHANTOM through the spring-damper system. The hole is static in the environment while the peg can be translated and rotated.

The haptic rendered model of the peg-in-hole insertion generating force data is constructed using the PointShell method. The PointShell of an object is the collection of all the points forming the surface of the object. A surface normal vector pointing inwards is assigned to every point on the PointShell to provide the Coulomb force direction [22, 23]. Fig. 3(a) illustrates the normal vectors of a PointShell. In the PointShell developed for the peg-in-hole insertion, the directions of the vectors assigned to singular points are not pre-determined as they depend on the normal of the contact surface (Fig. 3(b)). The directions are assigned when the peg and hole are in contact.

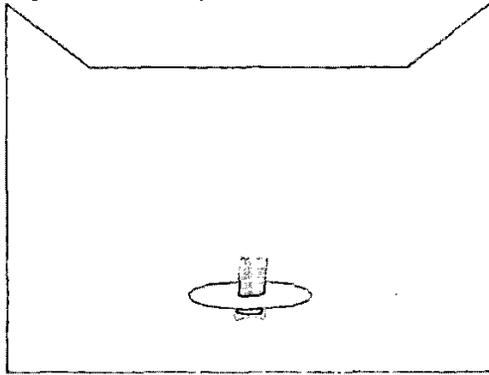


Fig.2 Peg-in-hole Insertion Virtual Environment

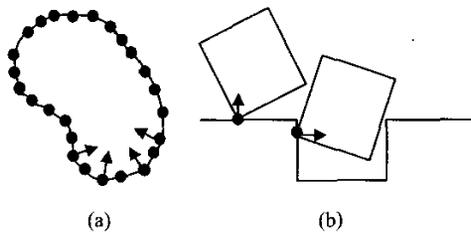


Fig.3 PointShell and Singular Points

The force generated at each point is the sum of the Coulomb force and the friction force exerted at that point, as shown in Fig. 4.

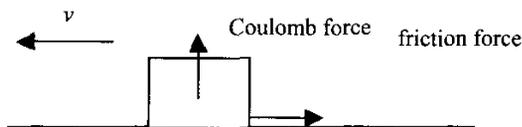


Fig.4 Coulomb force and friction force

The direction of the Coulomb force is perpendicular to the contact surface and points to the moving object. The magnitude of the Coulomb force generated at each point is calculated by

$$f_c = k*d + c*ad + b*v \quad (1)$$

where

d is the depth of the point in the contacting static object

ad is the accumulated depth during a continuous contact between the point and the static object

v is the velocity of the object and is calculated by the current *Depth* minus the last *Depth* divided by the sampling time

k is the stiffness coefficient

b is the damping coefficient

c is the coefficient for the accumulated depth

The direction of the friction force is along the contact surface and opposite to the moving direction. The magnitude of the friction force generated at each point is calculated by

$$f = \sigma*z \quad (2)$$

where

- z is the strain describing micromovements between the two objects, which is not allowed to exceed a small value called the breakaway distance z_{max} .
- σ is the stiffness relating force to strain

Assume x_i is a point fixed on the moving object, and y_i is an adhesion point on the static object as shown in Fig. 5. The following relationship is used to calculate z_i by

$$z_i = x_i - y_i$$

$$\begin{cases} y_i = x_i \mp z_{max}, & \text{if } |x_i - y_{i-1}| > z_{max} \\ y_i = y_{i-1}, & \text{otherwise} \end{cases} \quad (3)$$

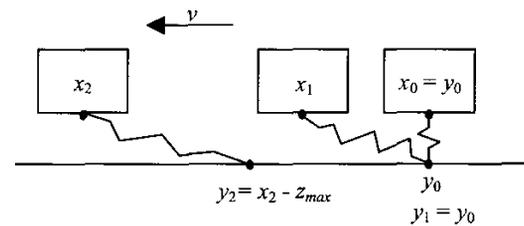


Fig. 5 The definition of the strain z

3. Skill Acquisition

A manipulation task consists of a sequence of basic skills. Identification of these basic skills and mapping them on to equivalent series of robot manipulation primitives, form the core of an algorithm for skill acquisition and transfer of those skills from human to a robotic manipulator. Such skill-based manipulation

is an effective way for a robotic manipulator to execute a complex task.

The basic skills are defined according to the contact state transition of a task, independent from the configuration of a manipulator [24]. In a virtual manipulation environment, the basic skills can be also identified by the contact states and state changes [25, 26]. Using this approach, the basic skills can be automatically extracted from the manipulation carried out in the virtual environment.

Skills can be classified at different levels according to difference between state changes. For example, the whole insertion progress can be divided into search and insertion phases. The search and insertion skills are two high level skills which result in critical state changes by driving the peg from the initial state to touch the hole and inserting peg after touching the hole respectively. Each high level skill can be divided into low level skills which result in minor state changes.

Two type of skills based on state changes can be used for assembly or manufacturing.

1. Skills based on both the current state and the next state
2. Skills based on the current state only

The first type of skills is learned during task sequence planning or trajectory optimization which finds the best state change sequence. In order to apply this skill, the next desired state or the method of choosing the next state should be known. State changes with the same current state but different next states might result in quite different output actions.

The second type of skills is acquired during performing a task with no obvious or fixed state change sequence. It is only based on the current state to simplify the skill learning process. It doesn't need to find an optimum state change sequence or follow a pre-defined state change sequence.

Some skill acquisition method can find the best sequence [27, 28] in the augmented state space formed by the current state and the next state. A point in the augmented state space represents a feasible state transition. However, this method requires a great deal of training data to explore the augmented state space and hence it is quite computing intensive for high dimension systems. For such reasons, this approach is not appropriate for on-line learning.

In this work, a simple method is employed to identify the optimal state change sequence for insertion phase. Initially different states are classified using a fuzzy neural network. This is achieved according to the forces/torque and translation along the Z-axis. The classification is then used to recognize the state change sequence for each training data file, in which the outputs are actions such as rotating the hole. The inconsistent or unintended actions such as movements of the peg during jamming should be identified and removed from the training data. The learning algorithm primarily learns the actions that result in a change of state.

The optimum sequence to perform the task is either identified from the training data or generated by

combining different sequences to a criterion such as the shortest time. If Sequence A takes shorter time from the initial state to a mid-state B but longer from state B to the final state than Sequence B, then the first part of Sequence A is combined with the second part of Sequence B to generate a sequence with shorter implementation time than both Sequences A and B. The actions or outputs recorded for each state change in the optimum sequence with different initial state are defined as the basic skills. For the round peg-in-hole insertion, the data can be mapped symmetrically to reduce the amount of training data needed.

In a practical insertion process, if a state is among or near any state in the optimal space, the recorded actions for these optimum sequences are employed to perform the task. On the other hand, if a state is not among or near any state in the optimum sequences stored in the skill library, a second type of skill generated through a fuzzy neural network is applied until a state among or near a state in the optimum space is produced (as shown in Fig. 6).

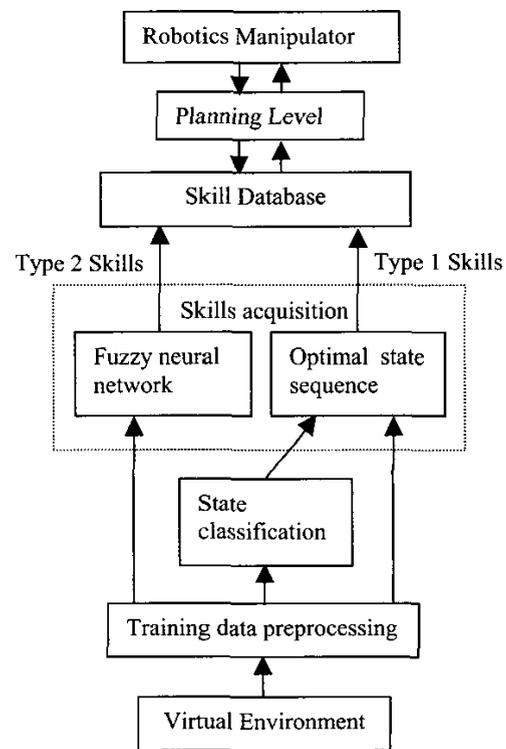


Fig.6 Skill Learning Structure

The experimental rig consists of a hole with two degrees of freedom (the pitch and yaw angles) controlled by two stepper motors, and a peg with one degree of freedom (the translation along the axis of the peg) controlled by a DC speed motor (see Fig. 7). These 3 DOF (Degrees of Freedom) are sufficient to study the insertion phase. The radius of the peg is 10mm and that of the hole is 10.05mm, so the clearance is 0.05mm.

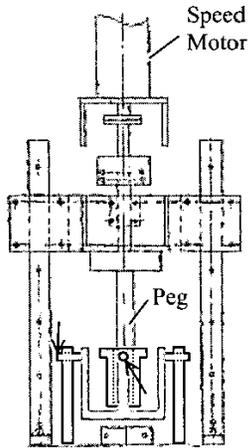


Fig.7 Real rig (Stepper motors are mounted on the axes marked by arrows)

M_x , M_y , z , Δz , $Step_x$ and $Step_y$ are illustrated in this diagram, where

- f_x , f_y , f_z are forces,
- M_x , M_y are torques,
- z is the translation along Z-axis,
- Δz is the change of the translation along Z-axis, and
- $Step_x$ and $Step_y$ are generated according to the acquired skills which indicate the steps taken by the step motor for turning the hole around x-axis and y-axis, respectively.

When the peg is jammed or some force threshold is reached, the actions are taken immediately, i.e., the peg stops moving down or even moves up a little bit and the step motors turn the hole accordingly. When the peg is not jammed, it is just moved down continuously. There are only 6 series actions taken in the insertion phase as shown in Fig. 8, which show the effectiveness of the method.

Some of the experimental results are illustrated in Fig. 8. The variation of 9 normalized series of f_x , f_y , f_z ,

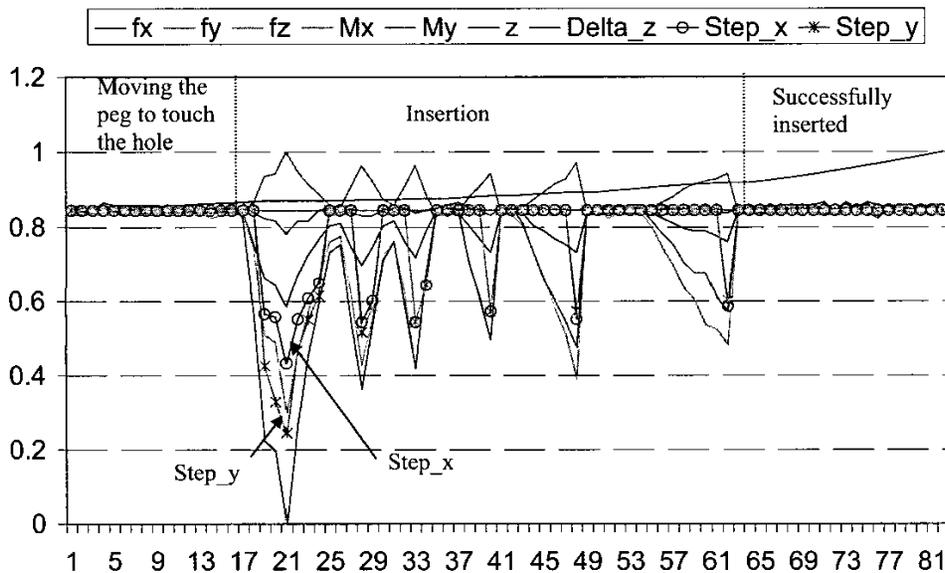


Fig.8 Results

4. Conclusion

The work conducted to study the transfer of manipulation skills from human to machine through a virtual environment with haptics feedback is reported. Broadly, the project has a generic scope that is novel and innovative. It explores how human motor manipulation skills can be replicated by a machine. The model identified for human psychomotor learning is emulated in the machine to achieve different stages of motor learning. The work also aims at providing a new insight into the nature of transfer of manipulation skills from human to machine.

The overall concept has been presented. The work at this stage has its focus on the peg-in-hole insertion

process. The concepts and methodologies developed for this application will be expanded in the next stage of the project towards more generic algorithms and models.

6. Acknowledgement

The support of ARC (Australian Research Council) and CRC (Cooperative Research Centre) for Intelligent Manufacturing and Systems Ltd for this project is acknowledged.

References

- [1] T. L. Perez and P. H. Winston, "LAMA: A language for automatic mechanical assembly," in

- Proc. Int. Joint. Conf. Artificial Intell., pp.321-333, 1977.
- [2] H. Matsubara, A. Okano, and H. Inoue, "Design and implementation of a task level robot language," *J. Robotics Soc. Japan*, vol. 3, no. 3, 1985.
- [3] S. J. Derby, "General robot arm simulation program (GRASP); Parts 1 and 2" in *ASME Comput. Engr. Conf.*, San Diego, pp139-154, 1982.
- [4] A. Naylor, I. Shao, R. Volz, R. Jungelas, P. Bixel, and K. Lloyd, "PROGRESS-a graphical robot programming system," in *Proc IEEE Int. Conf. Robotics Automat.*, 1987, pp.1282-1291.
- [5] B. Dufay and J. C. Latombe, "An approach to automatic robot programming based on inductive learning," *Int J Robotics Res.*, vol. 3, no.4, 1984.
- [6] P. D. Summers and D. D. Grossman, "SPROBE: An experimental system for programming robots by example," *Int J. Robotics Res.*, vol. 3, no. 1, 1984.
- [7] H. Asada and Y. Assari, "The direct teaching of tool manipulation skills via the impedance identification of human motions," in *Proc. IEEE Int. Conf. Robotics Automat.*, pp.1269-1274, 1988.
- [8] T. Sato and S. Hirai, "Language-aided robotic teleoperation system (LARTS) for advanced teleoperation," *IEEE Trans. Robotics Res.*, vol. 3, no.5 pp.476-481, 1987.
- [9] S. Suji, A. Morizono, and S. Kuroda, "Understanding a simple cartoon film by a computer vision system," in *Proc. Int. Joint. Conf. Artificial Intell.*, pp.609-610, 1977.
- [10] R. Thibadeau, "Artificial perception of actions," *Cognitive Science*, vol. 10, no.2 pp.117-149.
- [11] Y. Kuniyoshi, M. Inaba, and H. Inoue, "An approach to real time action understanding using robotic vision - Step 1: Generating state description sequences of an attention-getting object," in *Proc. Ann. Conf. Robotics Soc., Japan*, pp.435-436, 1987.
- [12] K. Ikeuchi and T. Suehiro, "Towards an assembly plan from observation," *Tech. Rep. CMU-CS-91-167*, School of Computer Science, Carnegie Mellon Univ, Pittsburgh, PA, 1991.
- [13] N. Haas, "Learning by ostentation for robotics assembly," in *Proc. SPIE*, 1991.
- [14] D. Sato, S. Yamada and M. Uchiyama, "Human skill analysis based on multi-sensory data," in *Proc. IEEE International workshop on Robot and Human Communication*, pp278-283, 1997.
- [15] Y. Kuniyoshi, M. Inaba, "Learning by Watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Transactions on Robotics and Automation*, Vol. 10, No.6, December 1994.
- [16] D. A. Pomerleau, *Neural Network Perception for Mobile Robot Guidance*, Kluwer Academic Publishers, 1993.
- [17] G. Z. Grudic, P.D. Lawrence, "Human-to-robot skill transfer using the SPORE approximation," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 2962-2967, 1996.
- [18] M. Kaiser, R. Dillmann, "Building elementary robot skills from human demonstration," in *Proceeding of the IEEE Int. conf. Robotics Auto.*, pp2700-2705, 1996.
- [19] D. A. Handleman, S.H. Lane, "Human-to-machine skill transfer through cooperative learning," *Intelligent Control Systems, Theory and Applications*, Edited by M.M. Gupta and N.K. Sinha, pp187-205, 1996.
- [20] F. A. Mussa-Ivaldi, N. Hogan, and E. Bizzi, "Neural, mechanical and geometric factors subserving arm posture in humans", *J. Neurosci.*, Vol. 5, pp.2732-43, 1985.
- [21] SensAble Technologies - Haptics Research. [online], available, <http://www.sensable.com/haptics/>
- [22] M. Renz, C. Preusche, M. Pötke, H.-P. Kriegel, G. Hirzinger. "Stable Haptic Interaction with Virtual Environments Using an Adapted Voxmap-PointShell Algorithm". *Proceedings of the Eurohaptics Conference*, Birmingham, UK, 2001
- [23] W. A. McNeely, K. D. Putterbaugh, J. J. Troy, "Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling", *Proc. ACM SIGGRAPH*, 401-408, 1999.
- [24] A. Nakamura; T. Ogasawara; T. Suehiro; H. Tsukune, "Skill-based back-projection for fine motion planning". *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems '96, IROS 96*, vol 2, pp 526 - 533.
- [25] J. Takamatsu; H. Kirnura; K. Ikeuchi, "Classifying contact states for recognizing human assembly task ". *Proceedings, 1999 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp: 177 -182, 1999.
- [26] H. Onda, H. Hirukawa, and K. Takase, "Assembly motion teaching system using position/force simulator - extracting a sequence of contact state transition". *Proceedings, 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems 95, 'Human Robot Interaction and Cooperative Robots'*, vol.1, pp: 9-16.
- [27] S. Lee and J. Chen, "Skill representation and acquisition", *Intelligent Systems for the 21st Century, IEEE International Conference on Systems, Man and Cybernetics, 1995*, vol.5, pp: 4325-4330.
- [28] S. Lee and S. Shimoji, "RCCN: radial basis competitive and cooperative network". *Proceedings. Fourth International Conference on Tools with Artificial Intelligence, TAI '92*, pp: 78-84, 1992.