The International Journal of **Medical Robotics** and **Computer Assisted Surgery**

THE INTERNATIONAL JOURNAL OF MEDICAL ROBOTICS AND COMPUTER ASSISTED SURGERY
Int J Med Robotics Comput Assist Surg 2012; **8**: 384–397.
Published online 18 May 2012 in Wiley Online Library (wileyonlinelibrary.com) **DOI:** 10.1002/rcs.1419

ORIGINAL ARTICLE

Selective automation and skill transfer in medical robotics: a demonstration on surgical knot-tying

Alois Knoll[1]*
Hermann Mayer[1]
Christoph Staub[1]
Robert Bauernschmitt[2]

[1] *Robotics and Embedded Systems, Technische Universität München, Garching, Germany*

[2] *Isar Kliniken München, Germany*

*Correspondence to: A. Knoll, Robotics and Embedded Systems, Technische Universität München, Boltzmannstr. 3, Garching 85748, Germany.
E-mail: knoll@in.tum.de

Accepted: 23 December 2011

Abstract

Background Transferring non-trivial human manipulation skills to robot systems is a challenging task. There have been a number of attempts to design research systems for skill transfer, but the level of the complexity of the actual skills transferable to the robot was rather limited, and delicate operations requiring a high dexterity and long action sequences with many sub-operations were impossible to transfer.

Methods A novel approach to human–machine skill transfer for multi-arm robot systems is presented. The methodology capitalizes on the metaphor of 'scaffolded learning', which has gained widespread acceptance in psychology. The main idea is to formalize the superior knowledge of a teacher in a certain way to generate support for a trainee. In our case, the scaffolding is constituted by abstract patterns, which facilitate the structuring and segmentation of information during 'learning by demonstration'. The actual skill generalization is then based on simulating fluid dynamics.

Results The approach has been successfully evaluated in the medical domain for the delicate task of automated knot-tying for suturing with standard surgical instruments and a realistic minimally invasive robotic surgery system. Copyright © 2012 John Wiley & Sons, Ltd.

Keywords medical robotics; automated surgery; learning by demonstration; imitation; scaffolding; skill transfer

Introduction

Medical robots are well on their way to becoming as important to the surgical process as industrial robots have become to manufacturing over the last 30 years. The surgeon is enabled by them to overcome several major barriers and limitations; the use of robotic systems will eventually lead to the successful performance of surgical procedures that would not have otherwise been feasible. The following are included by such barriers: (1) scaling of motion and precise manipulability [e.g. intrauterine fetal operations (1)], (2) access to hard to reach structures [e.g. by snake-like instruments for the emerging area of natural orifice transluminal endoscopic surgery (2)], (3) distance and teamwork during remote operations and (4) dexterity and speed by automated (sub-) sequences, which might enhance the surgeon's skills.

The latter issue of demonstrating the viability of selectively automated surgery for advanced interventional procedures is addressed by the overarching goal of the research work. From the surgeons' point of view,they will be allowed to move up in the 'hierarchy of controllers' by more 'intelligence' and adaptivity on the part of the robot. They will be freed by this development from performing tiring, tedious, low-level manual operations. However, despite the obvious advantages of automation, the surgeons will still remain in complete control – he or she will always be in command of the robot system. Clearly, this kind of situated behaviour intelligence can only be achieved when the robot system has access to all of the relevant parameters and background information necessary to perform an action partially or fully automatically. This is true both for (1) the relation between the situs and the robot (e.g. movement sequences and forces to be applied, expected results after a manipulation step and general knowledge about the organ being treated) and (2) the relation between the surgeon and the robot (e.g. what are the intentions of the surgeons, how do they express at what level of granularity they want to interact with the robot system and how can their hand movements be translated into movements of the instruments and the arms). In other words, advanced robots will only be realized if a new generation of surgical robots becomes an integral part of the 'information acquisition → planning → navigation/execution' loop with on-line access to all relevant information.

Moreover, these robots must be highly flexible with respect to the set of tasks they can perform, and they must be capable of continuously adapting to specific tasks as well as environmental conditions. The most complex of these operations is the task of knot-tying within the suturing process, which is the reason why we chose it to be our reference task (see the following text). Before detailing the methodological approach for the skill transfer in more detail, the concept of selective automation will be introduced. The key concept here is to break down a complex process into a sequence of tasks that can be performed with robots – either under classical master–slave control, partially autonomously or fully autonomously. With master–slave control, a given task is performed under direct control of the operator, that is, the robots follow the commands of the operator in a 1:1 fashion, but the operator has to make every decision on his own – down to the level of individual movements of robot segments. At the other extreme, in the case of full autonomy, a high-level command is issued by the operator, and how to best perform the task at hand is then autonomously decided by the robot system by using the information it has available about the environment – without any further intervention on the part of the operator; this is the classical 'sense → plan → act' loop. With partial autonomy, the operator is potentially involved in every aspect of the task execution. In this scenario, interference and re-instruction of the robot are possible at any time. All parameters can be set at the operator's discretion (e.g. the forces and velocities to be used and the starting points for movements).

Let it be imagined that the surgeon wants to tie a surgical knot (our reference task), which would be defined as one task. Then, in the full autonomous mode, it would suffice to simply tell the robot where to tie the knot, and all the rest would be left to the system to decide and execute. In partial autonomous mode, however, the knot would still be tied by the surgeon by moving the robots with joysticks, but certain movements would automatically be executed by the system (e.g. pulling the surgical thread after the knot structure was set up to achieve the right strain), and the surgeon would be prevented from performing movements into forbidden areas or with forces that are too strong.

In the next section, the methodology we have developed will be discussed for the whole chain of demonstrating selective automation for surgical interventions, in particular, the methods for task execution and skill transfer. We will start with a description of the task we were using as our benchmark example. There are quite a number of possible elementary tasks in surgery that would lend themselves to demonstrating the power of our approach, including cutting, holding, grasping and pulling out, and so on. However, the most complex in terms of required dexterity, precision, involvement of multiple manipulators, variability of desired outcomes as well as parameters to be measured and to be adapted is probably the task of tying a knot within the suturing process. Because of its complexity, this was chosen as our model task – if we can handle this specific task, it is expected to also handle all the other tasks that are normally needed in a typical operation process.

In this context, the term 'elementary' refers to tasks that comprise a separable and complete individual operation (e.g. one full knot). These individual sub-operations can be 'chained' together with other operations of the same kind (i.e. a sequence of knots forming a suture) or of a different kind (e.g. three cuts followed by a resection). It is noted that this task may also serve as an example of programming by demonstration (PbD) in other fields of robotics.

The principle of knot-tying in surgery is illustrated in the sequence of pictures in Figure 1, which shows the standard, textbook way of tying a common knot: first, the thread is grasped by the surgeon with his left hand (A), holding a pair of scissors in his right hand. Then, the thread is wound around the joint of the scissors a couple of times and grasps the other end of the thread with the scissors (B). Finally, he retracts the scissors (C). In a second (optional) step another single-winding knot is tied onto the first one. We observe that (1) this procedure is optimized for two effectors (i.e., the surgeon's hands) and that (2) the thread tension varies considerably during the procedure. Moreover, surgeons develop a delicate feeling for the forces they need to exert in all phases of this procedure.

So far, there has been only little research performed on the topic of automating the task of a surgical knot or other complex medical procedures of similar manipulation structure. A generalized version of a learned knot has been proposed by Kang in his PhD thesis (5). In his project, the task is performed with special knot-tying instruments (we are using general multi-purpose instruments). Robotic setups for knot-tying were proposed by Hynes *et al.* (6) and Wakamatsu *et al.* (7), but the task

DOI: 10.1002/rcs

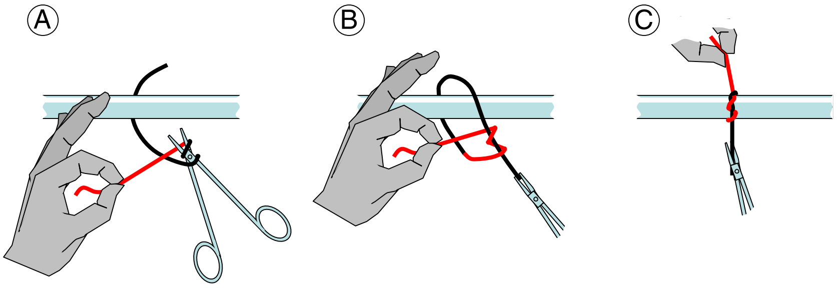


Figure 1. ***Manual knot-tying task*: subsequent steps of forming a surgical knot by a surgeon with the help of classical surgical instruments**

was not evaluated under realistic circumstances (i.e. small-scale knot performed under the restrictions of trocar kinematics and with original suture material). There is also some research on analysing the knot-tying task itself, focusing rather on surgical evaluation instead of automation (8,9).

The goal of the work described in the sequel is to show how a procedure of this complexity can be transferred to a robot system with minimal effort for an 'end user' and how this system can then execute the individual steps of the procedure.

Materials and methods

First, the telepresent system that was used for the experiments will be briefly described. For more details, we refer to (36). The slave manipulator of the system consists of four robots (cf. Figure 2). The robots can either be equipped with an endoscopic stereo camera or with minimally invasive instruments (3) on the basis of the 'daVinci' surgical instruments by Intuitive Surgical, Inc. (1266 Kifer Road, Building 101, Sunnyvale, CA). Optionally, one of the robots can be equipped with a stereoscopic endoscope instead of an instrument. The master console consists of two haptic displays ['Phantom' by Sensable Devices (Sensable, 181 Ballardvale Street, Wilmington, MA)] and a three-dimensional (3D) display, which is based on two polarized LCD screens and a semi-reflective mirror. As an additional input modality, foot switches are placed on the foot well of the console. With the switches, it is possible to hand over the control to different robotic arms. The so-called trocar kinematics is implemented by the control software of the system, that is, all instruments will move about a fixed fulcrum after insertion into the body or a ribcage mockup (4). In addition, translational forces applied to the instruments are measured by strain gauge sensors and are fed back to the operator through the haptic devices.

As a preview to the more detailed descriptions of experimental procedures described in the results section, the individual steps of the knot-tying procedure as performed by the system in automatic mode with three instruments is shown in Figure 3. After piercing the needle into the tissue by a human assistant, the needle with the thread is pulled out of the tissue with the left gripper (1) and wound about the right gripper (2). During this procedure, the loose end of the thread is strained by a third gripper, which operates force controlled. Now, the right gripper will be moved toward the loose end (3). To prevent damage to the thread, this movement has to be imitated by the left gripper. Finally, the end of the thread is pulled through the loop around the right gripper to complete the knot (4).

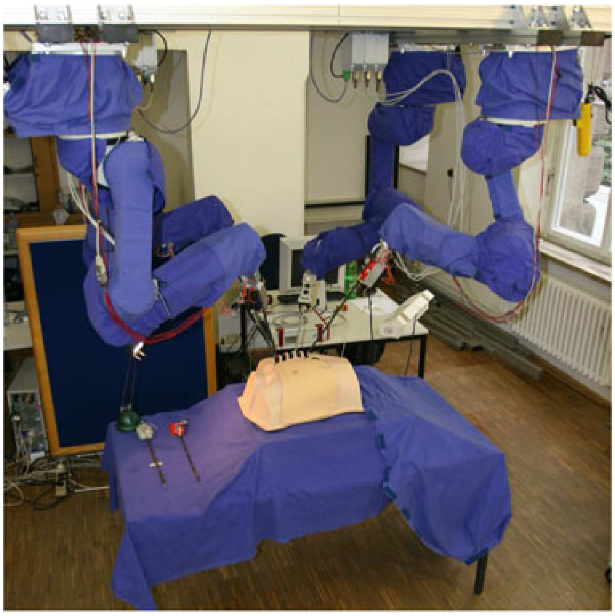


Figure 2. ***Hardware setup*: four ceiling-mounted robots with surgical instruments, either a stereoscopic endoscope or different surgical microgrippers. The minimally invasive instruments are augmented with force sensors. In-output is accomplished at a master console, comprising two force feedback devices**

Skill transfer

A common method for transferring human skills to a robot is 'PbD'. A good, more general overview that also emphasizes the importance of robot learning by demonstration, observation and imitation is given in a special issue of the transactions on system, man and cybernetics – part B (31). In this context, a PbD framework that allows for the extraction of relevant features of a given task and its generalization in different contexts for a humanoid robot is presented by Calinon *et al.* (32). A probability estimation is utilized to evaluate the relevance of the detected

DOI: 10.1002/rcs

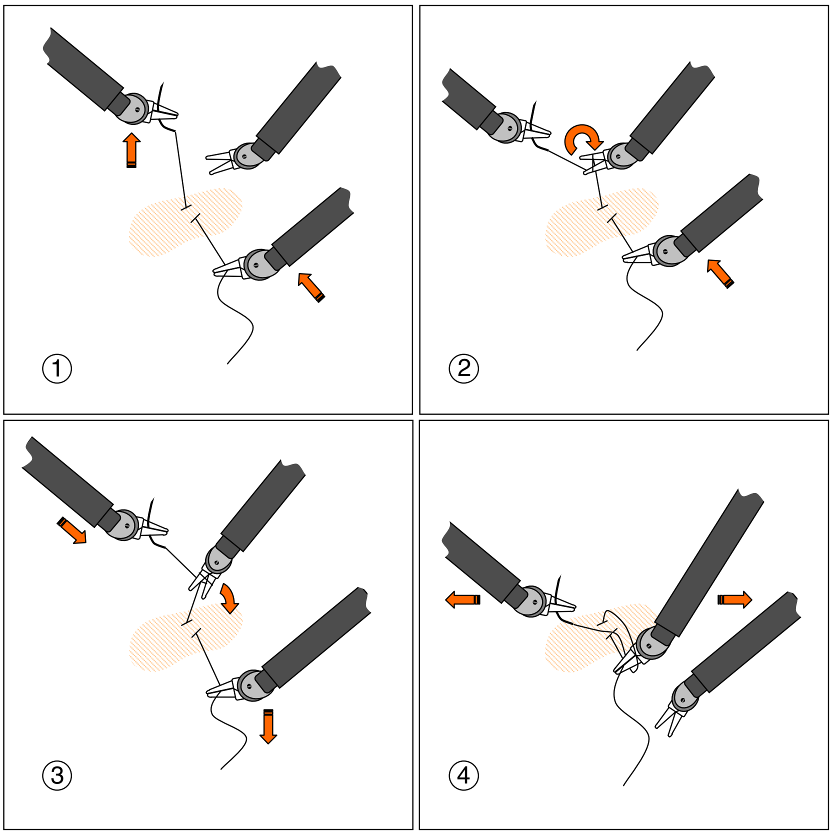


Figure 3. *Automatic tying of a knot*: four phases of an automatic tying procedure performed with three instruments

feature and a Gaussian mixture model to apply the task to a new environment. An architecture for PbD in (35) has been proposed by Zöllner *et al*. User demonstrations are decomposed into meaningful primitives, which are called 'elementary operation' in their work. A priori knowledge is included by means of defining a parameter space for each object. The generation of a hierarchy of subtasks ending up at basic primitives is induced by each action. Subtasks are arranged in a precedence graph, which is derived from previous demonstrations of the user. To decide whether a certain action can be applied, sets of pre-conditions and post-conditions are defined for each action. Given such a pre-condition, the corresponding precedence graph is serialized by the system, yielding miscellaneous alternatives for an execution plan.

In the following section, a special type of PbD is presented to transfer the manual skills of a surgeon to the robot system. First, the trajectories commanded by human operators with all geometric parameters, force sensor readings and effector commands were recorded. Then, this recorded procedure must be broken down into individual elementary actions. The latter are then transformed into chainable action primitives (cf. Figure 4). The following terms are defined: A *task* is a certain, well-defined description of how to solve a problem. A *demonstration* is an instance of this task performed by a human user. A *skill* is the instantiation of a task on a robotic system. Finally, *primitives* are meaningful, non-overlapping subsequences of a skill.

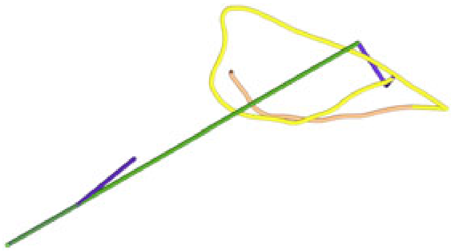


Figure 4. *Movement primitives*: each trajectory can be decomposed into a sequence of temporally non-overlapping primitives

In this terminology, to derive a robotic skill from user demonstrations, we proceed as follows:

1. *Smoothing and event detection* to find starting points of movement primitives.
2. *Decomposition* of the complete trajectory into meaningful primitives.
3. *Feature extraction* by means of template matching.
4. *Generalization and task instantiation* of skills by means of viscous flow simulation.

These steps will be detailed in the following subsections.

Smoothing and event detection

To observe knot-tying trajectories, a human surgeon worked with our experimental system in teleoperation mode, and performed the knot procedure using two hands. While the surgeon was tying the knot, the movements of the Phantom devices as well as other parameters such as forces were recorded, while the movements were directly executed by the two robot arms with their instruments.

The recordings typically started in a situation as shown in Figure 3(1), that is, the needle was anchored to the tissue. The results of these experiments were stored in a repository with numerous recordings of the trajectories of the left and right hand movements. It is shown in Figure 5 that the data acquired from user demonstrations were quite noisy. Moreover, the tremor of the surgeon (the *subject* performing the motion) is clearly shown. Smoothing this input data can be of some help, but information that is relevant in a given situation will also be typically deleted by naïve linear low pass filtering. For example, the correct picking up of the thread is represented by the 'edge' of the trajectory reaching out far to the left in Figure 5. If this edge peak were changed (even if the amplitude was reduced by just a millimetre), the thread would be missed by the robot performing the pickup operation later.

Therefore, before cleaning the data, the parts of the recorded data that are essential for the definition of the action primitives and those parts that can be smoothed out because they merely represent noise under the given task context must be extracted. This 'intelligent' smoothing has been implemented by spline approximation, with the constituting parts of the trajectory (such as the pickup edge) being selected as breakpoints.

To identify these breakpoints, however, additional cues are needed – this cannot be performed by looking at the spatial coordinates of the trajectory alone. Although in principle a number of additional modalities can be considered to provide more information (e.g. effector torque, optical analysis of distance between tissue and instrument tip and oral comments by the surgeon performing the operation), we mainly rely on the temporal dependencies within the trajectory and the state of the gripper. This methodology is based on research on the psychology of user interfaces: if a user moves the mouse pointer across the screen too fast, it is not possible to perform precise interactions with the user interface because the focus window of the eye is unable to achieve full focus and concentrate on the task (10). Furthermore, this observation is supported by Accot–Zhai's steering law (11), an enhancement of the well-known Fitts' law (12) for movements on constricted trajectories. In its original form, it is expressed as

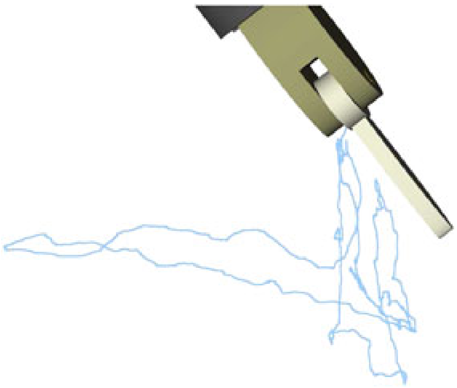


Figure 5. ***Knot-tying trajectory*: visualization of a typical recorded trajectory of the left gripper during knot-tying, as commanded by a human surgeon**

$$T_C = a + b\int_C \frac{ds}{W(s)} \tag{1}$$

where a and b are constants depending on the nature of the experiment and $W(s)$ is the width of the curved path C, at a certain location s. For clarification, this can be differentiated and rearranged to yield

$$\frac{ds}{dT} = \frac{W(s)}{b} \tag{2}$$

As can be seen from equation (2), the speed of the user moving on the trajectory is proportional to the width of the path. In other words, if the subject has to navigate on a narrowly constricted trajectory (i.e. performing a precise movement), it will take more time than an unconfined environment.

This is exactly the phenomenon on which we have based our algorithm to define interaction events without loss of critical features.

Decomposition

After smoothing the trajectory and transforming it into a spline representation, the next step is to decompose it into meaningful primitives exploiting the detected events as delimiters. As mentioned earlier, the decomposition of manipulation tasks into sensorimotor actions or motion primitives is a standard methodology to reduce complexity (13,14).

However, in our case, the selection of meaningful primitives is quite intricate because demonstrations are not normalized. Different demonstrations (by different humans) may be skewed, translated or rotated, and primitives may appear with totally different parameterizations. Therefore, procedures intended for normalized data cannot be used [e.g. (15,16)].

Instead, a special kind of a situated dialogue is utilized, where skills have to be demonstrated within the working domain and without requiring extrinsic knowledge. Systems based on situated dialogues between humans and machines have already proven effective in learning tasks (17,18). In our case, human–machine interaction is divided into two consecutive components: (1) an abstract description of the task and (2) its refinement through specific demonstrations. This strategy can be understood as a dialogue structure (Figure 7). A situated dialogue between a teacher and a trainee for the case of surgical knot-tying is shown in the left part of Figure 7. The first step is giving an *abstract description* of the very task that will be taught. This step is not situated because it is independent of the actual working environment. It can better be seen as a conceptual explanation, for example, like a person describing a procedure in general terms over the

DOI: 10.1002/rcs

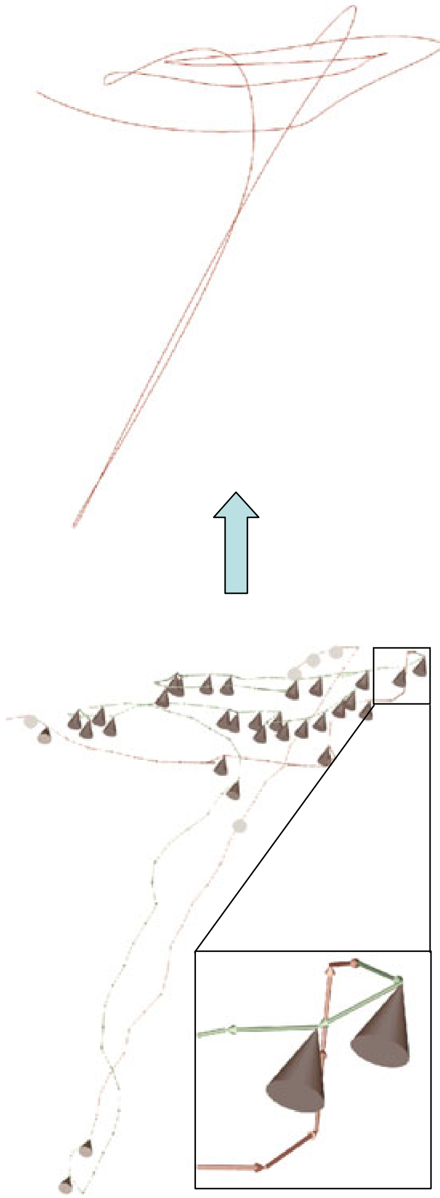


Figure 6. Spline approximation: Significant events are selected from the recorded trajectory (cones on left image). These points serve as breakpoints for a spline approximation of the trajectory (right image). Note that human tremor is suppressed, while all significant edges are preserved

telephone. Consequently, a skill will not be exclusively performed by the trainee on the basis of this description. The description rather serves as a pattern to draw attention to certain features and to distinguish which parts of the succeeding demonstrations will be important and which ones can be neglected. This is an important step towards learning based on a small number of user demonstrations (ideally just *one*). After giving this abstract description, the skill is concretized by the teacher by presenting instances in the application environment. Those will be used by the trainee to build an execution plan from the abstract description. This methodology is a specialty of situated learning and is called ‘scaffolding’, which traces back to the work of Vygotsky in the 1930s (19). The term scaffolding was first coined by Wood (20) and refers to the teacher giving assistance with respect to certain aspects of a task, which are beyond the capability of the trainee.

In our architecture, scaffolding is used to structure complex tasks by means of abstract patterns. This information cannot be derived by the system exclusively on the basis of user demonstrations, but it has to be communicated verbally (or in another modality) before the skill can be learned.

For this scaffolded learning scheme, the user has to specify a generalized version of the skill by means of a so-called *task template* (see Figure 8). No specific information (such as geometric features or forces) about the skill is contained but a sequence of basic actions is constituted by a task template . These actions are called *tasklets*, which define the atoms of the scaffolding scheme. Each tasklet starts and ends with a certain event (e.g. the turning point of the trajectory). In the previous section, how these events can be automatically detected in the demonstrated trajectories has been shown.

Once all of these events are identified, the optimal correspondence between the observed sequence of events and the task’s sequence of events (as constituted by the task template) must be elaborated by pattern matching. After that procedure, the user’s demonstration of a skill can be decomposed into a non-overlapping sequence of concrete primitives (Figure 4), which correspond to the abstract tasklets of the task. Only the relevant features of the primitives (defined in the corresponding tasklet, such as the start and end points) will be stored to generate a generic description of the skill, which can then be instantiated in a new environment [see (21)].

For our specific task of knot-tying, the following tasklets were pre-defined: linear movement, force controlled movement, two-dimensional (2D) movement and mutually synchronized movement of the two arms.

As mentioned earlier, tasklets do not contain any specific information such as starting points and absolute forces. These are provided by processing the information found in the concrete demonstrations. For example, it is stated only by the definition of the tasklet “2D movement” that a movement, which can be embedded into a 2D plane, can be found at that point in the task template. No geometric information about what these movements actually look like (for the 2D tasklet it could be a simple line but also a spiral winding motion in the plane) is contained by the tasklets. This information is only extracted when the tasklet is matched up with the corresponding primitive of the user demonstration. Clearly, any given trajectory can be decomposed into a sequence of 2D sections, which are instantiations of re-defined tasklets: every trajectory can be broken down into lines. However, the primitives are not desired to become too simple, that is, they must at least represent recurring actions with clearly discernible preconditions and post-conditions.

An important observation in this context is that the coarse movements of the distal extremities of humans (fingers, hands, feet) are typically constrained to two dimensions, even when they produce the most complex

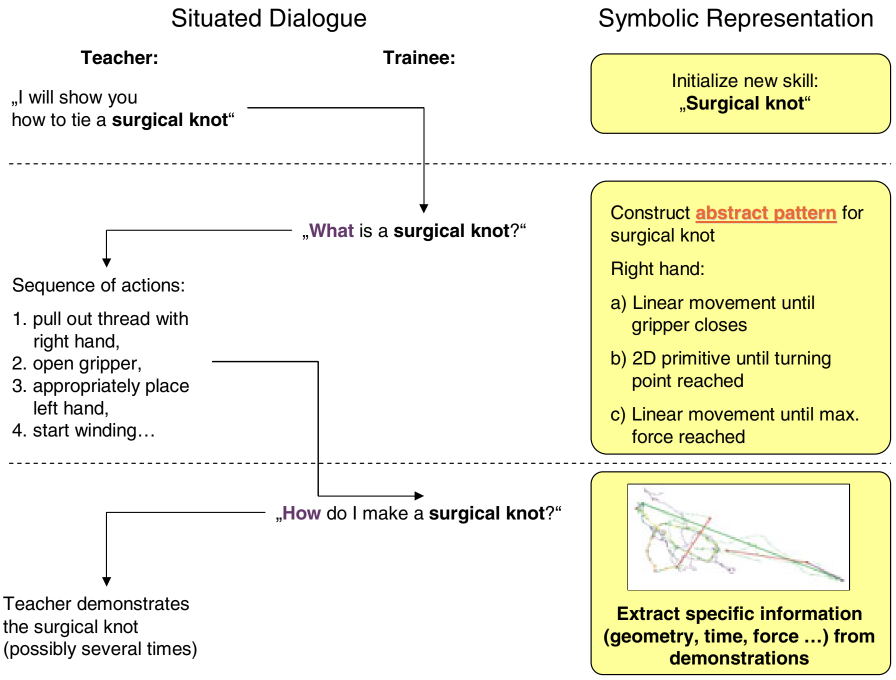

Situated Dialogue
Symbolic Representation
Teacher:
Trainee:
„I will show you how to tie a **surgical knot**“
Initialize new skill: „**Surgical knot**“
„**What** is a **surgical knot**?“
Construct abstract pattern for surgical knot
Right hand:
a) Linear movement until gripper closes
b) 2D primitive until turning point reached
c) Linear movement until max. force reached
Sequence of actions:
1. pull out thread with right hand,
2. open gripper,
3. appropriately place left hand,
4. start winding…
„**How** do I make a **surgical knot**?“
Teacher demonstrates the surgical knot (possibly several times)
Extract specific information (geometry, time, force …) from demonstrations

Figure 7. Scaffolding scheme as a situated dialogue between user and machine

behaviours: playing the piano or the guitar on a 2D keyboard or fingerboard and walking or even stair climbing can be normalized to two dimensions. These observations are also supported by recent research on learning sensorimotor behaviours (22,23). Therefore, the trajectory of any skill S is represented in the following way:

$$S = X_0 \cup T_1 P_1 \cup X_1 \cup T_2 P_2 \cup \ldots \cup T_n P_n \cup X_n = \bigcup_{i=1}^{n} T_i P_i \tag{3}$$

$$P_i = p_0 \circ \ldots \circ p_{n-1} \circ \varepsilon_i \tag{4}$$

$$P_i \cap P_j = \varnothing \ \forall i, j \in n; \quad i \neq j \tag{5}$$

where P_i is a 2D primitive, which implements a tasklet and is placed at its correct position by a rigid transformation T_i. Between the primitives, chunk sections X_i may occur, which have no relevance for the task and can be replaced by a direct connection of consecutive primitives.

To find a decomposition that is compliant with the predefined task, we have to check whether the skill is an instance of the corresponding task. This means we need to check whether the skill can be decomposed into a series of primitives complying with the sequence of tasklets. Equation (4) means that each primitive is a sequence of points delimited by a designated point p_n, at which a significant event (ε_i) occurs (i.e. $p_n = \varepsilon_i$). The test of whether primitives are instances of a certain task is performed by examining the events mentioned earlier. Besides using the events for resampling and smoothing the trajectory (with a high resolution in the cones of Figure 6), the event points are also important cues for selecting significant primitives. As initially mentioned, more events are detected in the concrete trajectory than are defined in the abstract task in the first scaffolding step (see Figure 10). Therefore, it is not possible to perform an unsupervised decomposition of the skill's trajectory exclusively on the basis of events – the information provided in the first scaffolding step is instrumental in decomposing the trajectory. For example, in the case of the knot-tying task, the first significant event is a movement cluster, depicted by a green circle in Figure 10. Therefore, this primitive is delimited by the beginning of the trajectory on the one side and by the first movement cluster event we come across when analysing the trajectory from left to right on the time line (cf. Figures 9 and 10). All other events occurring before this movement cluster (which we expected from the definition of the task template in the first step of the scaffolding) will be skipped. Let it be noted that the primitive starts at the beginning of the demonstration, although there is a time lag at the beginning of the first track in Figure 9. These pauses will be inserted in a subsequent step: after the instantiation of the tasklets, they are synchronized to be in line with the temporal specification of the task.

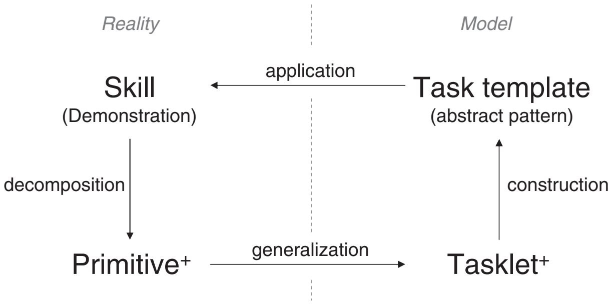


Figure 8. Terminology of our scaffolding skill transfer framework

This procedure is repeated until all detected events are processed. This part of the algorithm is depicted in the last row of Figure 13. If the algorithm is unable to find

DOI: 10.1002/rcs

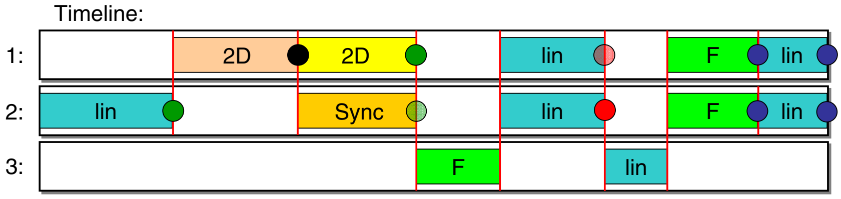


Figure 9. Knot-tying task as a sequence of tasklets. (lin, linear motion, 2D, two-dimensional primitive, F, force primitive)

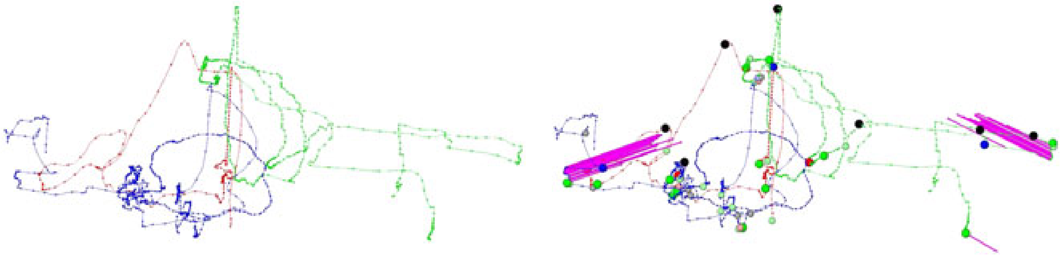


Figure 10. Original trajectory (left) and automatically detected events (right)

instantiations of all tasklets in the task template, the corresponding trajectory is rejected for not being a valid demonstration of the skill. Otherwise, the significant features (such as forces or the geometry of the trajectory) will be extracted from the detected primitives (≙instantiations of tasklets) in the next step.

Feature extraction

An applicable solution is produced by the learning procedure earlier, even after one single demonstration ('one-shot learning'). Once the primitives are derived from the trajectory and assigned to the corresponding tasklets defined in the first scaffolding step, the relevant information of the primitives can be extracted and stored in the knowledge base of the system.

This knowledge base consists of a general and a specific part. Information about the execution of primitives, which can be utilized in every task, is contained by the general part. For example, a trajectory generator is comprised by the general knowledge base to implement the straight line movements of linear motion primitives. Another example is the fluid simulation for the instantiation of 2D primitives (see the following text). This part of the knowledge base is fixed and will not be changed by user interactions. The only situation that requires a change of the general knowledge base is the addition of new tasklets. During normal operation, only the task-specific part of the knowledge base can be changed by the user. This comprises the definition of the task (in our case, the knot-tying task depicted in Figure 9) is comprised by this as well as extracted information from the demonstrations, which is necessary to carry out the task in real-world environments. The corresponding task-dependent information is extracted via selective data reduction. There is a special method of extracting data from primitives for each type of tasklet.

Two-dimensional movement. So far, the primitives, which are instantiations of this tasklet, are simply sequences of points in 3D space. As mentioned earlier, they are interpolation points of a spline representation of the trajectory. The central part of this approach is a plane-fitting function, which calculates an optimal plane for a given set of 3D points (i.e. the primitive). Optimality is defined by means of the minimum least squares method. The regression problem is stated as follows:

$$
\begin{aligned}
{}^{2D}z_i &= ax_i + by_i + c; \\
d_i &= z_i - {}^{2D}z_i; \quad \min\left(\sum_{i=0}^{n} d_i^2\right)
\end{aligned}
\tag{6}
$$

where (x_i, y_i, z_i) is the ith point of the corresponding primitive. Let it be noted that we are not using normal distances here, but directional distances, which are easier to obtain and provide reasonable results in our application domain.

The minimization is performed by linear regression. The return values of the function are the parameters of the planes a, b and c and the sum of squared distances. If the mean squared error exceeds a certain threshold (i.e. the points do not fit well on a 2D plane), the whole demonstration will be rejected for not being an instance of the pre-defined task. Otherwise, the plane parameters and the interpolation points of the spline are stored in the task-specific knowledge base for later instantiation of the task.

Linear movement. If a linear motion primitive was detected in the demonstration, it suffices to store the start and end point of the underlying trajectory. All other points in between can be omitted.

Force-controlled movement. As for the linear motion, the start and end point of the primitive is stored in the task-specific knowledge base. In addition, the maximum force vector is stored during the movement. This will be used later to control the instantiation of this tasklet.

Synchronized movement. As mentioned earlier, the synchronized movement tasklet can only exist if at least two independent trajectories have been recorded (e.g. left and right hand). Its purpose is to synchronize the movement section of one trajectory with the 2D movement occurring in another one [e.g. loop generation in Figure 3(2)]. The points of the corresponding 2D primitive of one trajectory are mapped onto the points of the synchronized movement primitive of another. The following transformation has been applied to describe this mapping:

$$\begin{pmatrix} x_1 \cdots x_n \\ y_1 \cdots y_n \\ z_1 \cdots z_n \\ 1 \cdots 1 \end{pmatrix} = A \begin{pmatrix} {}^{2D}x_1 \cdots {}^{2D}x_n \\ {}^{2D}y_1 \cdots {}^{2D}y_n \\ {}^{2D}z_1 \cdots {}^{2D}z_n \\ 1 \cdots 1 \end{pmatrix} \tag{7}$$

To calculate transformation A, each point in the trajectory of the referenced 2D primitive has to correspond to a point in the synchronized movement primitive. Therefore, the number of points n used for this procedure has to match for both primitives. Finding the transformation is formulated as the following singular value decomposition problem [cf. (24)]:

$$C = \frac{1}{n}\sum_{i=1}^{n} \left[\vec{p_i} - \bar{p}\right] \left[\overrightarrow{{}^{2D}p_i} - \overline{{}^{2D}p}\right]^T \overset{\text{svd}}{\rightarrow} C = USV^T \tag{8}$$

$$\text{where}\bar{p} = \frac{1}{n}\sum_{i=1}^{n} \vec{p_i};\ \overline{{}^{2D}p} = \frac{1}{n}\sum_{i=1}^{n} \overrightarrow{{}^{2D}p_i} \tag{9}$$

$\vec{p_i}$ is the ith point of the currently processed primitive, whereas $\overrightarrow{{}^{2D}p_i}$ is the ith point of the 2D primitive it is synchronized with. Let it be noted that both primitives are stored as splines, and therefore, both can be resampled with an equal number of n points. The eigenvalues s_1, s_2 and s_3 of matrix C^TC are contained by diagonal of S. The eigenvectors of CC^T and C^TC are contained by U and V, respectively. According to (24), the scaling factor f, the 2D rotation matrix R and the translation vector $\vec{t}$, which can be used to map ${}^{2D}p_i$ onto p_i, can be determined:

$$\begin{aligned} f = \frac{s_1 + s_2 \pm s_3}{\sigma_P^2} \text{ where } \sigma_P^2 \\ = \frac{1}{n}\sum_{i=1}^{n}\left(\overrightarrow{{}^{2D}p_i} - \overline{{}^{2D}p}\right)^2 \end{aligned} \tag{10}$$

$$R = UV^T \tag{11}$$

$$\vec{t} = \bar{p}_i - f \cdot R\overline{{}^{2D}p_i} \tag{12}$$

Let it be noted that the sign of s_3 depends on the determinant of C, when calculating f (in addition, the sign of the last row in U has to be adapted accordingly, when calculating R).

Generalization and task instantiation

Up to this point on our way from scaffolding to task instantiation, spline representations of the tasklets have been derived from user demonstrations. This is sufficient for reproducing a certain skill from equation (3) if all T_i are known. However, the direct application of splines may have some major shortcomings. The most important of these is that any shifting of the breakpoints, which may become necessary when adapting to a new environment, can lead to poor trajectories (25). Moreover, it is difficult to store temporal features such as velocity with splines because the dependency between the interpolation parameter and the arc length is non-linear. Therefore, for the instantiation of the primitives, the use of the dynamical systems known from fluid dynamics has been proposed.

So far, there has been only little research on dynamical systems for storage and generation of motion primitives. Dynamical systems were employed by Ijspeert (26) as generators for motion patterns to mimic locomotion of animals. A related approach was proposed by Okada (27). Attractors of dynamical systems were used to generate and stabilize walking movements of a humanoid robot. Both approaches operate on the joint level of motion generation, whereas our method generates trajectories in Cartesian space. Recently, there has also been research performed on analysing trajectories by means of dynamical systems. A method was proposed by Dixon (28) for segmenting primitives on the basis of linear dynamical systems. Although this can be used to segment and store motion patterns, the expressive power of the derived primitives is limited, and therefore they cannot be used for generalization.

Time-invariant dynamical systems, which lead to straightforward and stable solutions for movements in joint space, are used by each of the aforementioned approaches. However, by contrast, our goal is to provide trajectory generation in Cartesian space for rather complex motions (e.g. self-crossing trajectories). Therefore, a time-dependent system that can reproduce complex trajectories is needed. Such systems are applied in fluid dynamics, where they are used to simulate physical effects, for example, in a wind tunnel. Streamlines occurring in these environments are trajectories of particles in a fluid.

To the best of our knowledge, there has been no attempt to utilize this form of trajectory generation for robotic applications. For our approach, a dynamical system based on Navier–Stokes equations has been chosen. The behaviour of a viscous, incompressible fluid exposed to friction and external forces is described by these equations. The derivation of the equations can be found in various textbooks on fluid dynamics [e.g. (29)]. For our purposes, a simplified form of the equations with constant density has been chosen because we restrict our approach to incompressible flows:

$$\frac{\partial \rightarrow u}{\partial t} + \left(\vec{u} \cdot \nabla\right)\vec{u} + \nabla p = \nu \Delta \vec{u} + \vec{f} \tag{13}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{14}$$

∇ is the Nabla operator, and Δ is the Laplace operator: $\Delta = \nabla \cdot \nabla = \nabla^2$; $\vec{u}$ is the velocity of the fluid, ν its viscosity and $\vec{f}$ are external forces such as gravity. As mentioned earlier, the primitives we want to represent with this dynamical system are already in 2D space. Therefore, equations 13 and 14 can be instantiated to become [cf. (30)]:

$$\frac{\partial u}{\partial t}+\frac{\partial p}{\partial x}=\nu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)-\frac{\partial u^2}{\partial x}-\frac{\partial (uv)}{\partial y}+f_x \qquad (15)$$

$$\frac{\partial v}{\partial t}+\frac{\partial p}{\partial y}=\nu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)-\frac{\partial (uv)}{\partial x}-\frac{\partial v^2}{\partial y}+f_y \qquad (16)$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \qquad (17)$$

where u and v are velocities in x and y directions, respectively.

The equations are evaluated by means of finite differences within a rectangular area, which is subdivided into a grid of equally sized cells. Within these cells, the partial derivatives can be replaced by local difference quotients, for example, $\left[\frac{\partial u}{\partial x}\right]_{ij}\mapsto\frac{u_{ij}-u_{i-1j}}{d}$, where d is the length of each cell. A detailed description of this methodology can be found in (30).

Now, a demonstrated skill can be reproduced by reassembling the corresponding primitives with equation 3. Therefore, each primitive will be instantiated by a 2D fluid simulation. This can be achieved by sampling points from the spline representation of the corresponding primitive. In this case, equidistant sampling with length d is applied. There is no direct way to guarantee an arc length of d for spline $S(x)$ because Δx cannot be determined from $S(x+\Delta x)=S(x)+d$. This problem has been solved pragmatically by a binary search: Let Δx be an arbitrary initial value and $S(x)$ maps to a coordinate within the cell with velocities u_{ij} and v_{ij}. Then, we test for $S(x+\Delta x)$ lying within any adjacent cell. If it still lies in cell ij, $S(x+2\Delta x)$ is tried, if it even lies outside an adjacent cell, $S(x+0.5\Delta x)$ will be tried and so on (note that Δx refers to a distance and has nothing to do with the Laplace operator in equation 13).

Once we have sampled an applicable point S_x from the trajectory, the speed at this point can be determined with the help of the original trajectories, which are stored together with this primitive. Afterwards, the neighbouring values of u_{ij}, v_{ij}, u_{i+1j} and v_{ij+1} (see Figure 11) are interpolated. This means we calculate a preset for these velocities at time step t_n. All other velocities within the grid are derived from fluid simulation.

To illustrate this approach, think of throwing a particle into the stream, then it will be attracted to the trajectory of the underlying primitive. Because we know the position and orientation of the simulation grid from equation 3, suitable 3D points can be generated for a skill. Because the simulation is only refreshed at discrete points in time ($t_0<t_n<t_{\max}$), we have to interpolate again to obtain positions at arbitrary points in time. Fortunately, this works well even for tiny time steps. Therefore, positions can be sampled at the frequency of the controller of the robot (approximately 150 Hz). Therefore, our robots can be directly controlled with positions from fluid simulation.

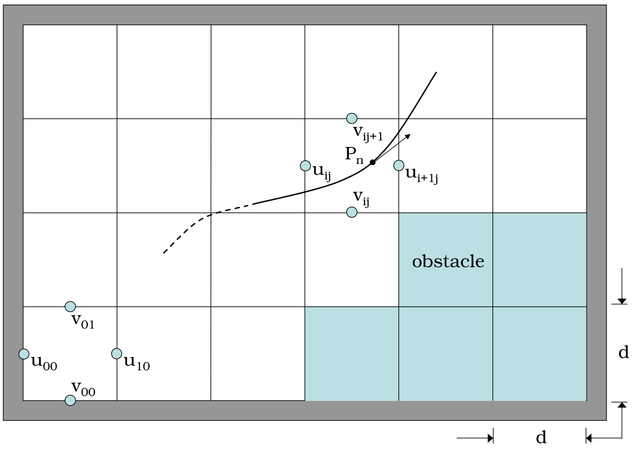


Figure 11. ***Fluid simulation*: the area of interest is discretized into equally sized cells. For clarity reasons, only a limited number of cells that would be too coarse for practical use is shown in this picture – our practical implementation usually works on a grid of at least 50 × 50 cells. Evaluation of velocities is not centred within a cell but distributed on a staggered grid to assert numeric stability**

Results

A short video of the system during operation can be found on the web[1] as well as an automated knot-tying sequence.[2]

- *Step 1*. Recording of demonstration data with a number of users.
- *Step 2*. Automatic extraction of primitives from these user-provided data sets.
- *Step 3*. Application of the respective task template to specific situations.

Overall, the results from these experiments were highly satisfactory, and the validity of both the methodologies for learning and for complex trajectory generation in the presence of obstacles has been fully justified.

The main difficulties were encountered in the application phase (Step 3 earlier), and they were due to imperfections in our hardware setup, which essentially has a very low repeatability (because of the compliance of the robot gantry and the flexible instrument shafts) and a low overall absolute accuracy because of uncalibrated robots. Clearly, both can be overcome by compensating offsets through (incremental) visual servoing, but this is beyond the scope of this work. Some specific comments follow, referring to the three steps above.

Recording phase

A sequence of images taken during a user demonstration of the knot-tying task is shown in Figure 12. After going through the demonstrations, data were collected for 30 knots.

Extraction phase

With these demonstration data sets, the primitives were generated according to the methodology described

[1] www6.in.tum.de/pub/Main/ResearchSfb453/system.avi
[2] www6.in.tum.de/pub/Main/ResearchSfb453/learned_knot.wmv

DOI: 10.1002/rcs

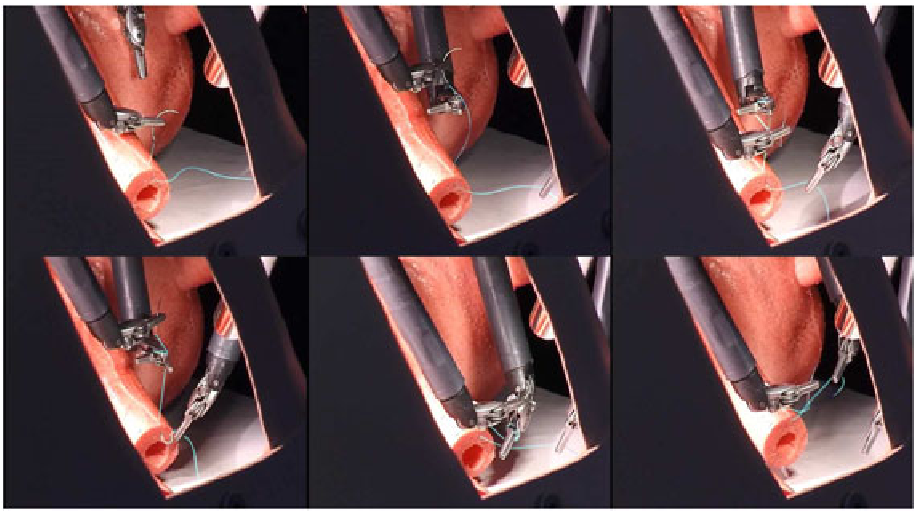
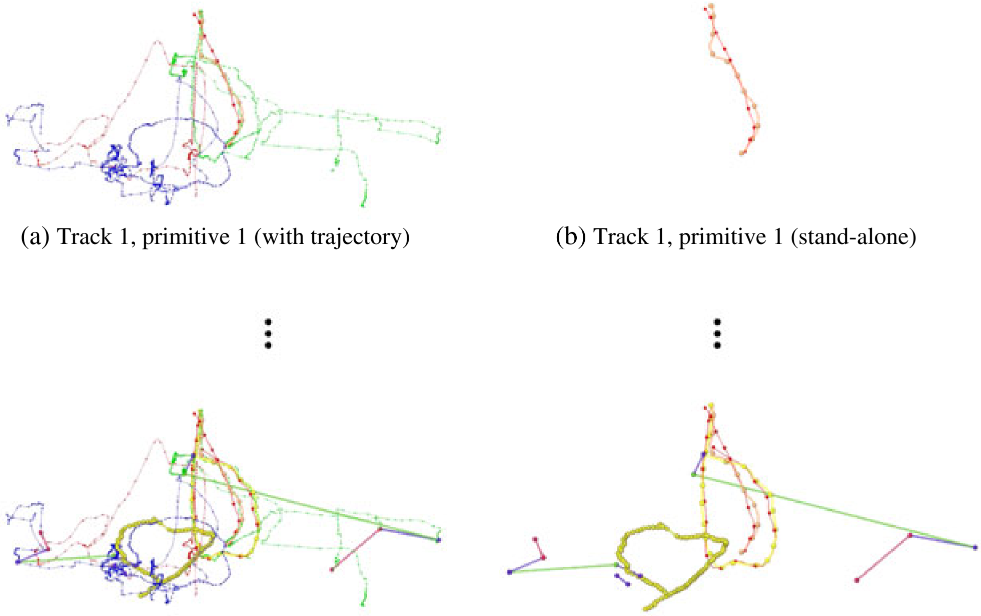


Figure 12. Subsequent stages of the knot-tying process during one user demonstration

earlier. At the beginning of this generation process, a number of these demonstration trajectories were rejected. This was because users did not stick to the abstract plan provided by the scaffolding framework. Typically, a technique for knot-tying was employed by such users, which was not compatible with the rough pattern defined by the scaffolding framework. However, after explaining the intended pattern to the users, the success rate of the demonstrations increased to almost 100%. By this, it is implied that in the future, users should be shown the scaffolding pattern in a cognitively adequate way before the demonstration or they should be allowed to change the scaffolding pattern by themselves. Nonetheless, to have a consistent data base for one specific task, all applications were based on one single task pattern. The resulting decomposition of a user-demonstrated trajectory is shown in Figure 13.

Application phase

Out of the 30 demonstrations, 15 were chosen at random to apply the corresponding skill to a new environment, that is, a setting in which we shifted the organ (in our case a cadaver heart) a few centimetres to the side to find out whether a task template could still be applied. This worked fine as far as the instantiation of the task template to the new geometry was concerned. There were, however, a number of adaptations needed to the fluid simulation before acceptable trajectories were generated. The first effect we observed was a deviation of the particle from the intended path. This was



(a) Track 1, primitive 1 (with trajectory)

(b) Track 1, primitive 1 (stand-alone)

⋮

⋮

(e) Track 2, primitive 6 (with trajectory)

(f) Track 2, primitive 6 (stand-alone)

Figure 13. Illustration of subsequent steps of the segmentation algorithm

due to the fact that the speed of the particle was not set appropriately, and it was not possible to stop the particle exactly at the right moment. Moreover, a particle that was travelling too fast was often drawn into a vortex, as depicted on the left side of Figure 14. This behaviour was easily suppressed by a calibration of the Reynolds number of the fluid simulation and by exciting the fluid for a longer period in the same place and by stretching the stirring to a path of points rather than just a single point. The problem of stopping the particle at the right time was solved by defining an ‘outflow’ condition at the point where the stirring path ended. To make the transitions smooth, the outflow is faded in by a sigmoid function at the end of the stirring. A visualization of the outflow and the correctly instantiated trajectory is shown in the area on the right of Figure 14.

Ideally, we would now be able to demonstrate a certain skill in one location and then apply it to an organ in a different location (which is still within the workspace of the robot arms). However, the distance offset we can actually achieve in our setup is relatively small. As mentioned before, this is due to a number of shortcomings in our experimental system: the absolute accuracy of the robot arms is limited to 2 mm (depending on the individual robot). In addition, the exact location of the base point of the robots is not known exactly. More importantly, in our setup, the coplanarity of the robots’ bases cannot be guaranteed. The major drawback with a high precision application, however, is due to the high elasticity (already ±15 mm for a small force) of the instrument shafts. This elasticity, which cannot be measured during an operation, makes it virtually impossible to move the tip of the instrument to a desired set point. Despite these shortcomings, we were able to run the skill resulting from 15 demonstrations at different points in the environment, provided there was only a limited displacement of the desired operation location.

Discussion

A novel approach has been presented for the transfer of complex sensorimotor human skills to multi-arm robot systems. Our approach is based on situated learning or more precisely, the scaffolding, which is an area of active research in psychology and cognitive sciences about human communication and teaching.

One of the elements of this approach is that the teacher and the trainee share the same environment and that the results can be generalized later to different situations and environments. Not only do they work in the same environment but they also have roughly the same sensorimotor capabilities for the interaction with the environment. This means that in principle, they can relate to each other’s mental concepts, affordances and sensorimotor contingencies – which is a basis for the mutual understanding during the scaffolding phases. Although in our current system, these concepts are static, it can easily be imagined that they may become dynamic and evolvable if there are adequate representations of these states on the robots’ side.

A trainee can only be successfully taught by a trainer if he is aware of the restricted capabilities of the trainee. Obviously, tasks that can be easily performed by humans are major challenges for robots. For example, rotating an object with the fingers of one hand is easy for adults, but because of the different kinematics, it is very difficult to mimic by artificial robot hands, even if they are multi-fingered. Moreover, there is also typically a lack of sensory information (e.g. tactile fingertip sensors comparable with human fingertips) and slow inflexible sensorimotor feedback loops.

From this point of view, scaffolding also comes in handy because these deficiencies, if made known to the teacher, can be integrated into the scaffolding phase – much like humans teach children in a different way if they know that they have not (yet) developed certain skills in their own individual development.

This is the reason, therefore, that the scaffolding can easily be adapted to a wide spectrum of capability pairs between teacher and trainee. In other words, this framework should be powerful enough to significantly increase the learning rate. On the other hand, it should be avoided that the trainee is asked too little; he should be forced to make his own efforts to generalize the teacher’s demonstrations.

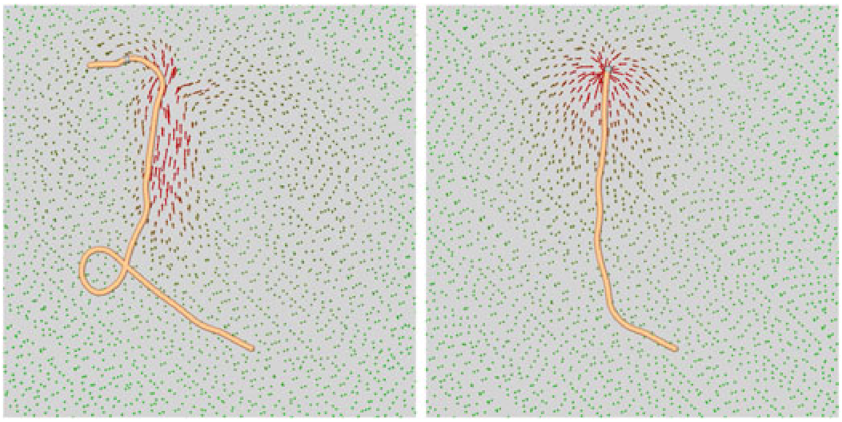


Figure 14. Different results of the fluid simulation for instantiation of the same tasklet when the speed of the particles are changed. Left: trajectory with particles travelling too fast, right: trajectory after speed adjustment

In the future, this skill transfer methodology will be extended to domains that are different from surgical knot-tying. This could be industrial robotic, household robotic or humanoid robotic scenarios of various kinds. An interesting neighbouring scenario is the handling of limp structures such ropes, cables and rubber seals. Here, the link to handling of surgical threads is obvious: the manipulated objects change their geometrical shape in an unpredictable way, and it is also difficult for humans to describe (and programme) the necessary skills – other than by demonstration. In this context, it would be highly desirable to integrate camera images for task-specific visual servoing. How this can be performed, for example, by verbally describing the expected image resulting from each step in the scaffolding is, however, a completely open question. There are also interesting links between our approach and the developments in the field of surgical process modelling (33) and surgical 'language' (34). For example, it would be highly interesting to explore how scaffolds could be constructed automatically from such process models and, conversely, how our methodology might serve as a representation for fine-grained descriptions of surgical interventions. These could then also be used for surgeons to discuss and compare individual aspects of interventions. Finally, in the upcoming 'pro-sumer market', that is, the production of small quantities of parts produced by the consumer at home, one could think of robot systems that assemble 3D printed parts. Clearly, this might be performed fully automatically if there are programmes written for controlling the assembly sequence. Alternatively, one could think of a more interactive process with the consumer instructing the robot system assembling products from a more generic set of parts. The construction processes would have to be adapted to a previously unknown or partly known shape of the product. In such a scenario, both the skill transfer and the fluid-based adaption of trajectories can be applied.

References

1. Harada K, Tsubouchi K, Chiba T, Fujie M. Micro manipulators for intrauterine fetal surgery in an open MRI. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005; 502–507.
2. Swanstrom L, Whiteford M, Khajanchee Y. Developing essential tools to enable transgastric surgery. *J Surg Endosc* 2008; **22**(3): 600–604.
3. Mayer H, Nagy I, Knoll A, Schirmbeck EU, Bauernschmitt R. Robotic system to evaluate force feedback in minimally invasive surgery. In *Proceedings of DETC, ASME Design Engineering Technical Conferences*, Salt Lake City, USA, 2004, nr. 57046 [CD ROM].
4. Mayer H, Nagy I, Knoll A. Kinematics and modeling of a system for robotic surgery. *On Advances in Robot Kinematics*. Kluwer Academic Publishers: Sestri Levante, Italy, 2004; 181–190.
5. Kang H. Robotic assisted suturing in minimally invasive surgery. Ph. D. Thesis, Rensselaer Polytechnic Institute, Troy, New York, 2002.
6. Hynes P, Dodds G, Wilkinson A. Uncalibrated visual-servoing of a dual-arm robot for MIS suturing. In *Proceedings of the IEEE International Conference on Biomedical Robotics and Biomechatronics*, Pisa, Italy, 2006; 204–209.
7. Wakamatsu H, Tsumaya A, Arai E, Hirai S. Manipulation planning for knotting/unknotting and tightly tying of deformable linear objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005; 2516–2521.
8. Kitagawa M, Okamura A, Bethea B, Gott V, Baumgartner W. Analysis of suture manipulation forces for teleoperation with force feedback. In *Proceedings of the 5th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Vol. **2488**, Dohi T, Kikinis R (eds). Lecture Notes in Computer Science, Springer-Verlag, 2002; 155–162.
9. Cao C, MacKenzie C, Payandeh S. Task and motion analyses in endoscopic surgery. In *Proceedings ASME Dynamic Systems and Control Division*, Atlanta, USA, 1996; 583–590.
10. Blackwell A, Jansen A, Marriott K. Restricted focus viewer: a tool for tracking visual attention. In *Theory and Application of Diagrams*, Vol. **1889**, Anderson M, Cheng P, Haarslev V (eds). Lecture Notes in Artificial Intelligence (LNAI), Springer-Verlag, 2000; 162–177.
11. Accot J, Zhai S. Beyond Fitts' law: models for trajectory-based HCI tasks. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, Atlanta, Georgia, USA, 1997; 295–302.
12. Fitts P. The information capacity of the human motor system in controlling the amplitude of movement. *J Exp Psychol* 1954; **47**: 381–391.
13. Kaiser M, Dillmann R. Building elementary skills from human demonstration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, USA, 1996; 2700–2705.
14. Schaal S, Peters J, Nakanishi J, Ijspeert A. Learning movement primitives. Springer Tracts in Advanced Robotics: International Symposium on Robotics Research (ISRR), Siena, Italy, 2004 rec-nr. 1805.
15. Reng L, Moeslund T, Granum E. Finding motion primitives in human body gestures. In *6th International Gesture Workshop*, Vol. **3881**, Gibet S, Courty N, Kamp J (eds). Lecture Notes in Artificial Intelligence (LNAI), Springer-Verlag, 2006; 133–144.
16. Sielhorst T, Blum T, Navab N. Synchronizing 3D movements for quantitative comparison and simultaneous visualization of actions. In *Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Vienna, Austria, 2005; 38–47.
17. Zhang J, Knoll A. Control architecture and experiment of a situated robot system for interactive assembly. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, D.C., USA, 2002; 3906–3910.
18. Kruijff G, Zender H, Jensfelt P, Christensen H. Situated dialogue and understanding spatial organization: knowing what is where and what you can do there. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*, Hatfield, United Kingdom, 2006; 328–333.
19. Vygotsky L. *Mind in Society: Development of Higher Psychological Processes* (14th edn edn). Harvard University Press: Cambridge, MA, USA, 1978.
20. Wood D, Bruner J, Ross G. The Role of tutoring in problem-solving. *J Child Psychol Psychiat* 1976; **17**: 89–100.
21. Mayer H, Burschka D, Knoll A, Braun EU, Lange R, Bauernschmitt R. Human-machine skill transfer extended by a scaffolding framework. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, California, USA, 2008; 2866–2871.
22. Aflalo T, Graziano M. Possible origins of the complex topographic organization of motor cortex: reduction of a multidimensional space onto a two-dimensional array. *J Neurosci* 2006; **26**(23): 6288–6297. Stanford University's HighWire Press
23. Amirikian B, Georgopoulos A. Modular organization of directionally tuned cells in the motor cortex: is there a short-range order? *Proc Natl Acad Sci USA* 2003; **100**(21): 12474–12479.
24. Umeyama S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans Pattern Anal Mach Intell* 1991; **13**(4): 373–380.
25. Schaal S, Ijspeert A, Billard A. Computational approaches to motor learning by imitation. *Philos T Roy Soc B* 2003; **358**(1431): 537–547.
26. Ijspeert A, Crespi A, Cabelguen J. Simulation and robotic studies of salamander locomotion. Applying neurobiological principles to the control of locomotion in robots. *Neuroinformatics* 2005; **3**(3): 171–196.
27. Okada M, Osato K, Nakamura Y. Motion emergency of humanoid robots by an attractor design of a nonlinear dynamics. In

Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005; 18–23.

28. Dixon K, Khosla P. Trajectory representation using sequenced linear dynamical systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005; 3925–3930.
29. Chung T. *Computational Fluid Dynamics*. Cambridge University Press: New York, USA, 2002.
30. Griebel T, Dornseifer T, Neunhoeffer T. Numerical simulation in fluid dynamics. A practical introduction. SIAM Monographs on Mathematical Modeling and Computation no. 3, Society for Industrial and Applied Mathematics, Philadelphia, USA, 1997.
31. Demiris Y, Billard A. Special issue on robot learning by observation, demonstration, and imitation. *IEEE Trans Syst Man Cy B* 2007; **37**(2): 254–255.
32. Calinon S, Guenter F, Billard A. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Trans Syst Man Cy B* 2007; **37**(2): 286–298.
33. Neumuth T, Jannin P, Strauss G, Meixensberger J, Burgert O. Validation of knowledge acquisition for surgical process models. *J Am Med Inform Assoc* 2009; **16**(1): 7280.
34. Reiley C, Lin H, Varadarajan B, Khudanpur S, Yuh D, Hager G. Automatic Recognition Using Statistical Modeling for Capturing Variability of Surgical Motions, Medicine Meets Virtual Reality Conference (MMVR), Long Beach, CA, USA, Jan. 2008.
35. Zöllner R, Pardowitz M, Knoop S, Dillmann R. Towards cognitive robots: building hierarchical task representation of manipulations from human demonstrations. Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, April 2005.
36. Mayer H, Nagy I, Knoll A, Braun EU, Bauernschmitt R, Lange R. Haptic feedback in a telepresence system for endoscopic heart surgery. *Presence-Teleop Virt* 2007; **16**(5): 459–470. MIT-Press

DOI: 10.1002/rcs