Remote minimally invasive surgery – haptic feedback and selective automation in medical robotics

Christoph Staub[a,*], Keita Ono[b], Hermann Mayer[a], Alois Knoll[a], Heinz Ulbrich[b]
and Robert Bauernschmitt[c]
[a]*Robotics and Embedded Systems, Technische Universität München, Munich, Germany*
[b]*Institute of Applied Mechanics, Technische Universität München, Munich, Germany*
[c]*German Heart Center Munich, Clinic for Cardiovascular Surgery, Munich, Germany*

Abstract. The automation of recurrent tasks and force feedback are complex problems in medical robotics. We present a novel approach that extends human-machine skill-transfer by a scaffolding framework. It assumes a consolidated working environment for both, the trainee and the trainer. The trainer provides hints and cues in a basic structure which is already understood by the learner. In this work, the scaffolding is constituted by abstract patterns, which facilitate the structuring and segmentation of information during "Learning by Demonstration" (LbD). With this concept, the concrete example of knot-tying for suturing is exemplified and evaluated. During the evaluation, most problems and failures arose due to intrinsic system imprecisions of the medical robot system. These inaccuracies were then improved by the visual guidance of the surgical instruments. While the benefits of force feedback in telesurgery has already been demonstrated and measured forces are also used during task learning, the transmission of signals between the operator console and the robot system over long-distances or across-network remote connections is still a challenge due to time-delay. Especially during incision processes with a scalpel into tissue, a delayed force feedback yields to an unpredictable force perception at the operator-side and can harm the tissue which the robot is interacting with. We propose a XFEM-based incision force prediction algorithm that simulates the incision contact-forces in real-time and compensates the delayed force sensor readings. A realistic 4-arm system for minimally invasive robotic heart surgery is used as a platform for the research.

Keywords: Medical robotics, telesurgery, time-delay compensation, skill transfer, visual guidance, automated surgery

1. Introduction

Since its introduction in the early 1980s minimally invasive surgery (MIS) has become a promising option for an ever-growing number of procedures. On one hand, significantly reduced hospitalization time through the quicker recovery of patients, less pain and collateral trauma as well as lower infection rates [20]

are the most important benefits to patients. On the other hand, surgeons have to cope with increasingly complex working conditions. Long instruments, unfamiliar and sometimes awkward to operate for the surgeon, are used through small incisions or ports in the body of the patient to perform the intervention. In contrast to the conventional open surgery, visual access to the internal operating scenery is not feasible. This constrains the visual perception to an endoscopic view without an intuitive depth perception or hand-eye coordina-

*Corresponding author. E-mail: staub@in.tum.de.

tion. Reduced degrees of freedom (DoF) introduced by the instruments and the trocar poses serious navigation coordination problems inside the limited workspace.

An important step in developing MIS and to overcome the drawbacks are telemanipulation systems, which have attracted the research community in recent years [31]. The systems are especially designed to enable delicate maneuvers with high surgical precision. The surgeon no longer controls the instruments directly, but they are controlled by special input devices with a Cartesian user interface without any chopstick effect (inversion of the moving direction). However, long-distance interventions suffer still from time-delays in communication channels which may cause synchronization between operator movements and contact force perception to be lost.

Although medical robots are well on their way to become as important to the surgical process as industrial robots have become to manufacturing within the last 30 years, most researchers and surgeons agree that the lack of haptic feedback [19] and the fact that recent medical robots are still "unintelligent" machines [13] prevent them from being in major applications in hospitals. More intelligent systems and adaptivity on the part of the robot will allow surgeons to move up in the "hierarchy of controllers". This development would free them from performing tiring, tedious, low-level manual operations (e.g. knot-tying).

In this work, we focus on the automation of recurring tasks in minimally invasive surgery and on surgical teleoperation over delayed networks. We have chosen robotic knot-tying as a reference application for human-machine skill transfer, since it combines many difficulties occurring in automation such as the handling of limp objects. So far, only a few research projects have been devoted on this particular task of automating a surgical knot. Kang has proposed a generalized version of a taught knot in his PhD thesis [11]. In his project the task is performed with special knot-tying instruments, while we are using multi-purpose instruments. Hynes et al. [9] and Wakamatsu et al. [33] proposed robotic setups for knot-tying, but did not evaluate the task under realistic circumstances, yet (i.e. small-scale knot performed under the restrictions of trocar kinematics and with original suture material). There exist also some works on analyzing the knot-tying task itself, with focus rather on surgical evaluation instead of automation [4, 12]. Problems that arise from intrinsic system imprecisions are compensated by visual instrument guidance.

Surgical teleoperation involves with many different types of tissue interactions such as tissue cutting with a scalpel or needle piercing. A natural display of forces is important for the skill transfer, since forces are utilized in the learning process during user demonstrations. For this reason, time-delays in the communication channel are highly critical to the learning rate of the system and also for the manipulation quality in general. Therefore, we interest in enabling a stable and realistic perception rendering to the operators when data communication is imperfect or delayed. The synchronization between hand movements of the operator and the force perception is then lost if a large time-delay is present [6]. In terms of control theory, this matter can be solved using a state-estimator in order to predict the behavior of the process with less actual feedback signals from physical sensors [28]. For a telepresence system, the state-estimator is comparable to a contact model which calculates the contact force between a tool at the end-effector and an interacted object in real-time and substitutes it from the delayed force signal. The contact model can be derived depending on the type of contact, using the principle of multi-body mechanics for solid objects [8] or with finite elements for a contact with deformable objects [26].

2. Robotic system

In strong collaboration with the German Heart Center Munich, a telemanipulation system for robotic heart surgery was developed. The system went through several phases of successive refinement, whereas the robotic hardware as well as the usability of the master console was improved (see [15]). In the current version, it consists of four small industrial robots (Mitsubishi 6SL^{TM}) with 6 DoF each, which are mounted on a aluminum gantry (cf. Fig. 1(a)). The ceiling mounted arrangement was chosen after an extensive evaluation with surgeons [3] while permits good access to the operating table, although all four robots are used. The robots can either be equipped with minimally invasive instruments or an endoscopic camera. Through the aid of a magnetic clutch the instruments can be interchanged quickly for better handling. The mechanism will also disengage the instruments if forces beyond a certain level are exerted and prevents damage in case of a severe collision. The surgical instruments itself are based on the commercial "EndoWrist^{TM}" instruments by Intuitive Surgical, Inc. and offer 3 DoF. A micro-
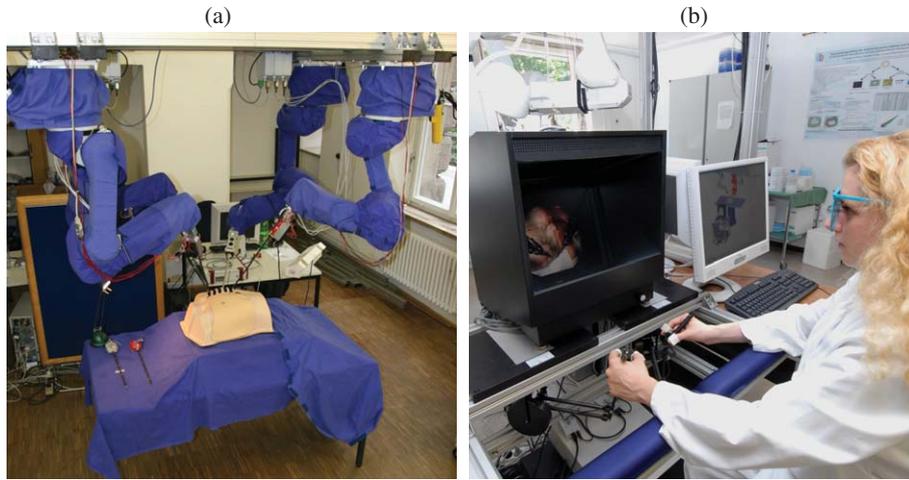
(a) (b)

Fig. 1. Hardware setup: Four ceiling mounted robots with surgical instruments, either a stereoscopic endoscope or different surgical micro-grippers. The minimally invasive instruments are augmented with force sensors. In-/output is accomplished at a master console, comprising two force feedback devices. Visual feedback is accomplished by a 3D-display with polarization technique.

gripper at the distal end of the shaft can be rotated, and adaption of pitch and yaw angles is possible. All movable parts of the gripper are driven by steel wires, connected to small servo motors by an Oldham coupling. Since the *z*-axis of the instrument and the robot flange align, we obtain 8 DoF. The master console of the system (the workstation of the surgeon) comprises two haptic devices ("PHANToM^{TM}Premium 1.5" by Sensable Devices) which are mounted upside down on a quickly reconfigurable aluminum frame. This arrangement yields to less constricted flexibility of the stylus pen. The position and orientation of each manipulator are controlled by moving the stylus pen that simulates the hand posture and feel of conventional surgical tools (cf. Fig. 1(b)). A switch is used to open and close the micro-gripper at the distal end of the instrument shaft. Forces exerting on the instruments are measured by strain gauge sensors and fed back by means of the haptic devices. The system is equipped with a 3D endoscope providing two separate optical (fiberglass) channels and two synchronized CCD cameras. The stereo information of the camera system is visualized at the operators side by a 3D-display. The display is based on two polarized LCD screens and a semi-reflective mirror. Therefore, the surgeon has to wear polarized lenses in order to get a 3D impression. As an additional input modality, foot switches are placed at the footwell of the console. The switches allow for the hand-over of the control to different robotic arms or can be programmed freely with

several system functions. The control software of the system implements so-called trocar kinematics, i.e. all instruments will move about a fixed fulcrum after insertion into the body or a ribcage mockup by geometrical calculations [17].

3. Selective automation

For recent surgical robotic systems, the surgeon mainly performs the visual servoing by himself to establish hand-eye coordination and executes tasks as he observes the scene and reacts by controlling the instruments movements. Using this classical 1 : 1 master-slave control forces the surgeon to make every decision on his own. To move him up in the hierarchy of controllers, the system has to be highly flexible to autonomously offer or complete different actions in an unknown and changing environment. The key concept of selective automation is to divide a complex task into a sequence of smaller tasks that can be performed by robots — either under classical master-slave control, partly autonomously or fully autonomously. While fully autonomy revokes the detailed control of a given high level command, partial autonomy involves the surgeon in every aspect of the task execution. In this scenario, interference and re-instruction of the robot are possible at any time. Clearly, this kind of situated behavior intelligence can only be achieved when the robot system has access to all of the relevant param-

eters and background information. This is true both for (i) the kinematic relation between the situs and the robot (e.g. movement sequences and forces to be applied) and (ii) the relation between the surgeon and the robot (e.g. what are the intentions of the surgeons, scene understanding).

3.1. Skill transfer extended by a scaffolding framework

Recurrent tasks in the context of medical procedures include pushing, holding, cutting or suturing. The most complex of these operations is the task of knot-tying during the suturing process, which is the reason why we chose it to be our reference task. Before discussing the method we have developed for the transfer of skills from a human operator to a robot system, we illustrate the principle of surgical knot-tying (see depiction of Fig. 2). Like the approaches of other authors (e.g. [25]), the presented skill transfer framework is based on the well-known technological paradigm of splitting up complex tasks into smaller parts. Transferring this paradigm to robotic learning means means to assemble task-specific skills from a set of sensorimotor primitives [14]. We observed that the illustration depicted in Fig. 2 is optimized for two effectors (i.e., the surgeon's hands), but is not necessarily the best decomposition of the task in order to transfer it to a technical system. Therefore, we started the skill transfer with the recording of trajectories (left and right hand, also cp. section 3.1.1.) commanded by a human operator, including the force sensor readings after an initial situation as depicted in Fig. 2(1) was reached. Then, a certain skill demonstration by the user must be broken down into individual elementary actions to derive primitives that

can be reassemble by the system to an autonomously executable task. At this point, we define the following terms that are used in the next sections:

A *task* Φ is a certain, well-defined description of how to solve a problem and can be decomposed into so-called *tasklets* ϕ (the plus sign in ϕ^+ indicates that a task consists of at least one tasklet). Given this template, the demonstration of the user is decomposed into a sequence of *primitives*, which corresponds to the decomposition of the task into tasklets as it is provided by the scaffolding framework. While a *skill* (the user's demonstration) is the instantiation of a certain task performed by a human user and primitives are instantiations of tasklets. Like every task, a tasklet is a transformation between pre- and post-conditions of an environment. Here, an environment is a set of objects Γ and a set of properties Θ of these objects. The properties are defined quite generally and can refer to geometric quantities (like rigid translations or distances) as well as forces or torques exerted to the object. A task is a transformation of an environment at t_0 into an altered environment at t_n, which will be expressed by $(\Gamma_0, \Theta_0) \overset{\Phi}{\longmapsto} (\Gamma_n, \Theta_n)$. The transition $\overset{\Phi}{\longmapsto}$ constitutes the task Φ. In many cases Γ_0 will be equal to Γ_n as long as the set of objects remains unchanged. The set will be changed, for example, if objects are destroyed or new objects are generated out of existing ones. For clarification, refer to the following description of the well known peg-in-a-hole task: The environment is defined by the set of objects $\Gamma_{ph} = \{peg, hole\}$ with Θ_0 comprising the rigid transform T_{ph}, which denotes the position of the peg relative to the hole. Therefore, the peg-in-a-hole task can be expressed as $(\Gamma_{ph}, \{T_{ph}\}) \overset{\Phi}{\longmapsto} (\Gamma_{ph}, \{I\})$, i.e. the goal is to align the peg with the hole until their centers are matched, and
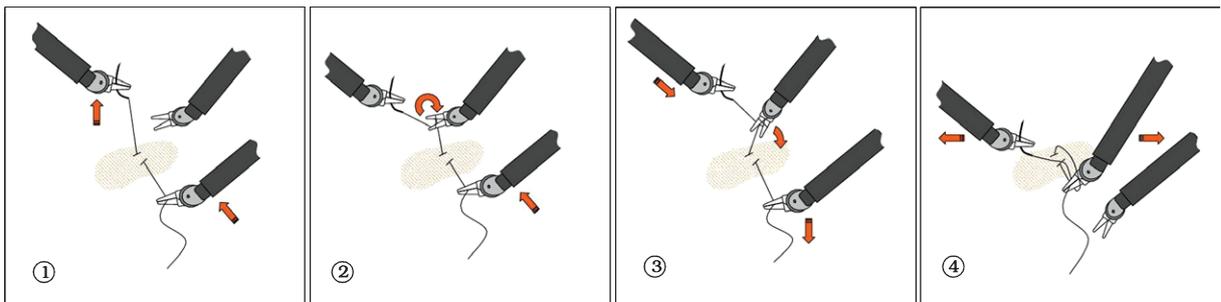

Fig. 2. Knot-tying procedure: After piercing, the needle with the thread is pulled out of the tissue with the left gripper (1) and wound about the right gripper (2). During this procedure, the loose end of the thread is retained by an assistant. This facilitates grasping of the end by the right gripper (3). Finally, the end of the thread is pulled through the loop around the right gripper in order to finalize the knot (4).

thus, the transformation between peg and hole finally constitutes a unit matrix I. Task definitions provide the system with abstract information about a task, which is going to be demonstrated by the user. Providing this kind of hints is the very essence of scaffolding as it is used in our skill-transfer architecture.

Given by the definitions introduced above, a tasklet will be denoted by a corresponding transition: $(\Gamma_a, \Theta_a) \overset{\phi}{\longmapsto} (\Gamma_b, \Theta_b)$, where $\overset{\phi}{\longmapsto}$ is the transition of tasklet ϕ. Therefore, a task is a set of tasklets, which met the following requirements:

$$\exists\, t_b \le tn : (\Gamma_0, \Theta_0) \overset{\phi}{\longmapsto} (\Gamma_b, \Theta_b)$$

$$\exists\, t_a \ge t0 : (\Gamma_a, \Theta_a) \overset{\phi}{\longmapsto} (\Gamma_n, \Theta_n)$$

$$\left\{ \overset{\Phi}{\longmapsto} \right\} \in \left\{ \overset{\phi}{\longmapsto} \right\}^+ \tag{1}$$

The first requirement determines the pre-condition of task Φ being also a pre-condition of at least one tasklet (the same accounts to the post-condition in the second requirement). The last statement refers to the transitive hull of the tasklet transitions: The transition of task Φ has to be included in the transitive hull in order to let these tasklets be a valid dissection of the task. For a complete description of the task, the transitions of the tasklets have to be ordered appropriately. Therefore, we need to extend our definition of a task by three different relations. Those are irreflexive, transitive relations defined on the set of tasklets $\phi^0 \ldots \phi^m$. The most important is the precedence relation "<", i.e. $\phi_0 < \phi_1$ self-evidently means tasklet ϕ_0 has to be completed before tasklet ϕ_1 starts. In addition, we need a synchronization relation "=". Accordingly, $\phi_0 = \phi_1$ indicates that the tasklets have to be carried out at exactly the same starting time, and also implies that execution times of both tasklets are equal. For integrity reasons we also define an exclusion relation "$\neq$", which prevents tasklets from being executed at the same time. An application example for this type of relation would be a situation where two manipulators cannot move to the same place at the same time, but the corresponding tasklets are not ordered by any precedence. Putting it all together we get the following **extended definition of a task**:

$$\Phi = \left\{ \phi^*, <, =, \neq \right\} \tag{2}$$

While the transition of a task Φ is determined in quite an abstract manner (transform environment (Γ_0, Θ_0) into environment (Γ_n, Θ_n)), we will define four con-crete types of tasklets: Linear motion, 2D motion, force controlled motion and synchronized motion. Those four will suffice to construct our application example of surgical knot tying. A **linear motion** is constituted by the following tasklet:

$$(\Gamma_a, \{T_a\}) \overset{\phi_{\text{lin}}}{\longmapsto} (\Gamma_b, \{T_b\}) \tag{3}$$

Γ_a contains at least the manipulator, which is going to carry out the movement. In the pre-condition, the manipulator is placed at posture T_a and should finally reach T_b in the post-condition. Both postures are interconnected by a linear movement regarding the translational part. Rotations are calculated by a spherical linear interpolation based on quaternions [27]. A more general version of this tasklet is the **2D motion**:

$$(\Gamma_a, \{T_a\}) \overset{\phi_{\text{2D}}}{\longmapsto} (\Gamma_b, \{T_b\}) \tag{4}$$

where both postures are interpolated by an arbitrary 2D spline, which interconnects the 3D coordinates of T_b and T_b. Again, rotations are handled by spherical linear interpolation. Note that this definition poses no further restrictions on the spline, except for embedding into a plane. The next tasklet, the **force restricted motion** is basically a linear motion:

$$(\Gamma_a, \{T_a, F_a\}) \overset{\phi_{\text{F}}}{\longmapsto} (\Gamma_b, \{T_b, F_b\}) \tag{5}$$

The difference is that the motion will be stopped before T_b is reached, if a certain force F_b is exceeded. In this case, the rest of the rotational motion will be carried out in place. The last tasklet we want to introduce here is the **synchronized motion**. This one is special, because, as a pre-condition, it requires another tasklet ϕ_s to exist, which is not a synchronized motion tasklet:

$$(\Gamma_a, \{T_a, \phi_s\}) \overset{\phi_{\text{syn}}}{\longmapsto} (\Gamma, \Theta) \tag{6}$$

The post-condition is arbitrary since the purpose of this tasklet is not directed towards a certain state of the environment, but it is correlated with the trajectory of another tasklet ϕ_s. This correlation can be realized by an affine bijection.

Different combinations of these tasklets can be used to form a task. In the upper left corner of Fig. 3(b) the task of surgical knot-tying is depicted (performed by three arms $\Rightarrow$ three tracks). Each tasklet is displayed by a colored rectangle. "2D" stands for a 2D primitive, "lin" for a linear motion, "F" for a force controlled

$$(\Gamma_0, \Theta_0) \overset{\Phi}{\longmapsto} (\Gamma_n, \Theta_n)$$

reassemble

$$(\Gamma_a, \Theta_a) \overset{\phi^+}{\longmapsto} (\Gamma_b, \Theta_b) \longleftarrow \text{generalize} \longleftarrow \text{Primitive}^+$$

Demonstrations

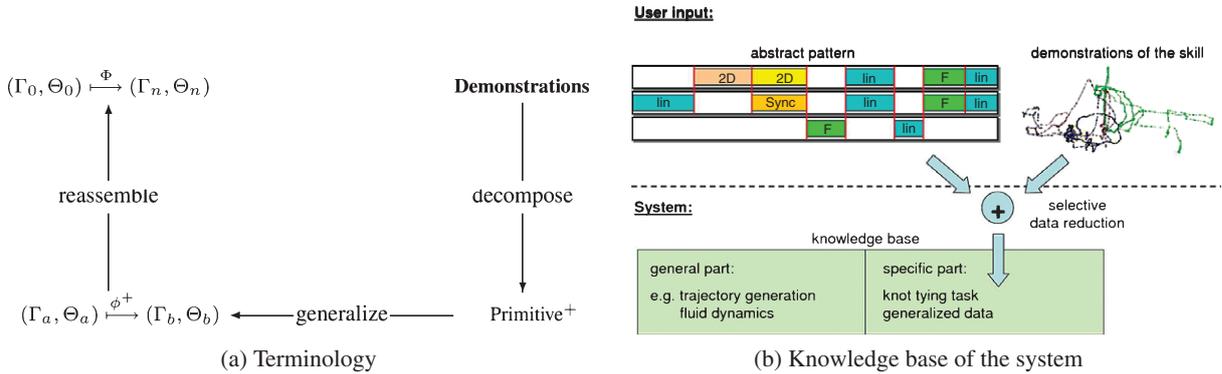decompose

(a) Terminology

(b) Knowledge base of the system

Fig. 3. (a) Summarizes the methodology of our scaffolding framework, while (b) depicts the system knowledge base.

primitive and "Sync" for a synchronized motion. The vertical red lines denote synchronization points. After all, it is possible to construct a broad variety of different tasks out of only four tasklets as they are defined above.

3.1.1. Feature extraction

In order to extract the relevant features from user demonstrations, the corresponding trajectories are pre-processed by an unsupervised event detection algorithm, which identifies certain states of the environment as they are defined in the Θ-part of the tasklets. Those states are used to match tasklets against the actual demonstration in order to derive corresponding primitives. Since the recorded data suffers from human tremor and is quiet noisy, the first step will be smoothing input data. Simple low pass filtering will typically also delete information which is relevant in a given situation. For instance, the "edge" of the trajectory reaching out far to the left in Fig. 4(a) represents the correct picking up of the thread. If this edge is smoothed out

even just for millimeters, the thread will be missed. Therefore, we have to be sure which parts of the trajectory are necessary for further processing and which parts can be neglected without consequences. We have implemented this feature by means of spline approximation, where important parts of the trajectory (like the pickup edge) are chosen as breakpoints. Therefore, we mainly rely on the temporal dependencies within the trajectory and the state of the gripper. The former is used to identify points where the subject moves rather slowly, which indicates a need for increased precision. The latter provides important information about those points in space where the subject was interacting with the environment by grasping objects (e.g. the thread). On the left side of Fig. 4(a), all locations where the subject remained within a radius of 1.5 mm for more than 0.2 sec are marked with a cone. This cone has the same orientation as the gripper which produced the trajectory. The closure of the gripper is indicated by the color of the trajectory, which changes from green

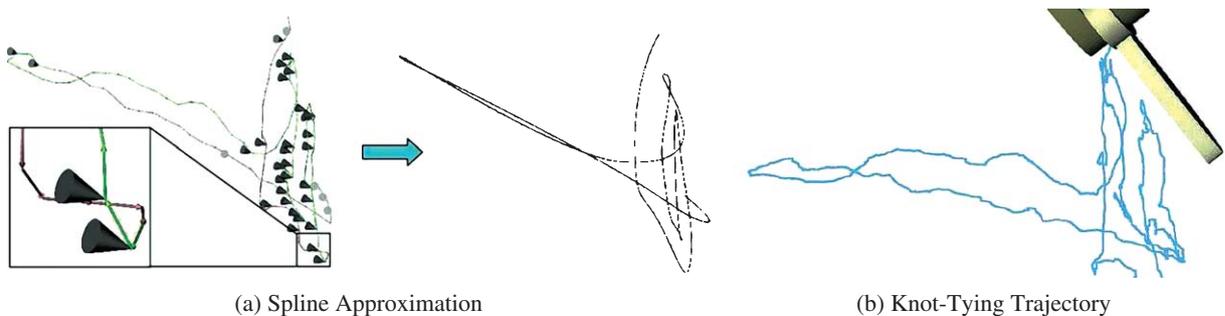

(a) Spline Approximation

(b) Knot-Tying Trajectory

Fig. 4. (a) 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. Fig. 3(a) summarizes the methodology of our scaffolding framework.(b) Visualizes a typical trajectory recorded from the left gripper during knot-tying, as commanded by a human surgeon.

to red. An important observation is, that there are no cones on the way to the pick up position and back. This indicates that the corresponding part of the trajectory is not important and therefore a distinct smoothing can be applied.

The extraction algorithm itself can be seen as a finite state machine, whose states are defined by the tasklets. This finite automaton processes the user input (demonstration) and checks whether it was an instance of the task definition or not. After each state change (each recognition of a tasklet), the automaton separates a primitive from the input and produces a corresponding output. For example, if an instantiation of a force controlled movement tasklet was detected in the input, the algorithm will extract the maximum force from the demonstration. This information is saved in order to instantiate the tasklets in a new environment. For example, once all primitives are derived from the demonstration, the relevant information is stored in the knowledge base of the system. The structure of the knowledge base and the procedure of inserting information is depicted in Fig. 3(b).

The knowledge base consists of a general and a specific part. The general part contains information about the execution of primitives, which can be utilized in every task. For example, the general knowledge base comprises a trajectory generator in order to implement the straight line movements of linear motion primitives. Another example is a fluid simulation for the instantiation of 2D primitives [16]. This part of the knowledge base is hard-wired and will not be changed by user interactions. The only situation, which requires an alteration of the general knowledge base is the extension of the system by new tasklets. This has to be effected by adjusting the source code of the system. During normal operation, the user can only change the task-specific part of the knowledge base. This contains the definition of the task (in our case, surgical knot-tying with three robotic arms) and extracted information from the demonstrations, which is necessary to actually instantiate the task in real world environments. The corresponding task-dependent information is extracted by selective data reduction of the primitives. There is a special method of data extraction for each type of tasklet:

(1) **2D movement**: After the pre-processing mentioned above, the points constituting the primitives are interpolation points of a spline representation of the trajectory. The central

part of feature extraction for 2D movements is an optimization algorithm, 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:

$$^{2D}z_i = ax_i + by_i + c; d_i = z_i - {}^{2D}z_i; Min\left(\sum_{i=0}^{n} d_i^2\right) \tag{7}$$

where (x_i, y_i, z_i) is the ith point of the corresponding primitive. The minimization is actually performed by linear regression. If the mean squared error exceeds a certain threshold (i.e. the points does not fit well to a 2D plane) the whole demonstration will be rejected for not being an instance of the predefined task.

(2) **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.

(3) **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 during the movement is extracted. This will be used later to control the execution of the instantiation of this tasklet.

(4) **Synchronized movement**: The synchronized movement tasklet can only exist in connection with a 2D tasklet. The points of the corresponding 2D primitive are mapped onto the points of the synchronized movement primitive. We have applied an affine transformation to describe this mapping:

$$\begin{pmatrix} x_1 \cdots x_n \\ y_1 \cdots y_n \\ z_1 \cdots z_n \\ 1 \cdots 1 \end{pmatrix} = A \begin{pmatrix} {}^{2D}x_1 \cdots {}^{2D}x_n \\ {}^{2D}y_1 \cdots {}^{2D}y_n \\ {}^{2D}z_1 \cdots {}^{2D}z_n \\ 1 \cdots 1 \end{pmatrix} \tag{8}$$

In order 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

affine transformation is formulated as the following singular value decomposition problem (cf. [32]):

$$C = \frac{1}{n} \sum_{i=1}^{n} \left[\vec{p_i} - \overline{p} \right] \left[\overrightarrow{^{2D}p_i} - \overline{^{2D}p} \right]^T \xrightarrow{svd} C = USV^T$$

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

Variable $\vec{p_i}$ is the i-th point of the currently processed primitive, while $\overrightarrow{^{2D}p_i}$ is the i-th point of the 2D primitive the former is synchronized with. Note that both primitives are stored as splines, and therefore, both can be resampled with an equal number of n points. The diagonal of S contains the eigenvalues s_1, s_2 of matrix $C^T C$. Matrices U and V contain the eigenvectors of CC^T and $C^T C$, respectively. By means of these results, we can determine 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:

$$f = \frac{s_1 + s_2}{\sigma_P} \quad \text{where} \quad \sigma_P = \frac{1}{n} \sum_{j=1}^{n} \left(\overrightarrow{^{2D}p_i} - \overline{^{2D}p} \right)$$

$$(10)$$

$$R = f \cdot UV \quad (11)$$

$$\vec{t} = \overline{p_i} - f \cdot R^{\overline{2D}p_i} \quad (12)$$

3.1.2. Experiments

The knot-tying task was demonstrated with the system depicted in Fig. 1(a). Three grippers are employed to perform the knot. During the data acquisition phase, two of them are controlled by the input instruments, one is autonomously controlled and keeps the loose end of the surgical thread under tension (in order to preserve a well-defined position). Afterwards, the matching algorithm of the scaffolding framework is employed to derive primitives from the demonstrated trajectory. On the left side of Fig. 5 the originally recorded trajectories of the grippers are depicted. On the right side of the figure the state of detection after all primitives have been detected is shown. Once the relevant information of a valid demonstration is stored in the knowledge base, the corresponding task can be instantiated in a new environment.

However, skill application did not produce a knot in all cases (success rate was 67% for twelve trials). The main difficulties were encountered in the application phase of the learned knot-tying procedure, and they were mainly due to imperfections in our hardware setup, which essentially has a very low repeatability (due to the compliance of the robot gantry and the flexible instrument shafts) and a low overall absolute accuracy. A crucial issue is that the absolute accuracy of the robot arms is limited to 2 mm (depending on the individual robot). Also the mechanical fit of the carbon-fiber shafts is quite large of about 15 mm. Even a carefully calibrated system cannot compensate the elasticity during operation and makes it virtually impossible to move the tip of an instrument to a desired position in space with a sufficient high precision.

A major issue for successfully transferring human skills to a technical system and automatically instantiate those tasks is a detailed understanding about the scene that is shared by the trainer and the trainee. This involves knowledge about the geometrical location and orientation of objects (e.g., the involved surgical instruments), as well as realistic contact forces between tissue and instruments with a minimum time-delay in communication. Both aspects are considered in the next sections.
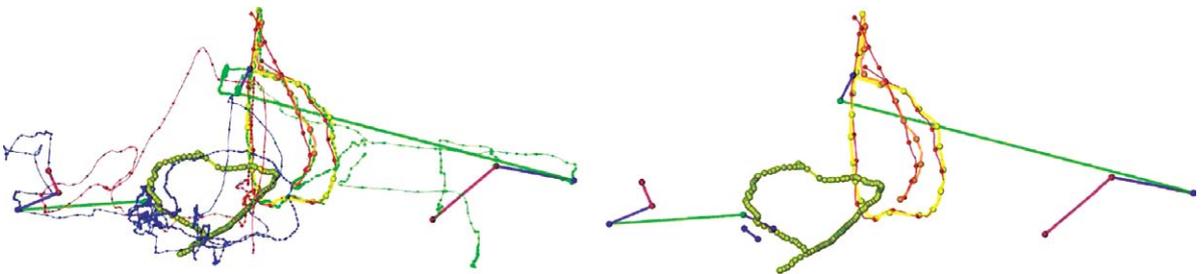

Fig. 5. Extracted primitives with original trajectory (left) and by themselves (right).

3.2. Visual guidance

Although we calibrated the system introduced in Section 2 carefully, the above mentioned inherent imprecisions cannot be determined with an satisfying accuracy. In particular, the transformation $^{F_2}_I T$ that follows from the aberration of the carbon fiber shaft of the instrument (cp. Fig. 6(a)) cannot be minimized to a satisfying level. Visual servoing is a popular approach to guide a robotic appendage (i.e. a surgical instrument) using visual feedback from a camera system. In general, visual servoing can roughly be divided into two categories: Position-based visual servoing control (PBVS), in which a Cartesian coordinate is estimated from image measurements and image-based visual servoing (IBVS) approaches, which seek to extract features directly from a series of images. The tracking of image features is not considered in this work, but is treated separately in [30]. For this work, a simple color segmentation was used to identify the surgical tool and a needle. The accuracy of image-based methods for static positioning tasks is less sensitive to calibration than PBVS [7], as the positioning error is directly reduced in image space. However, a practical difficulty during the alignment of surgical instruments with a desired position in space lies in the fact that the instrument is not necessarily in the field of view of the camera and therewith no image-features can be extracted. In order to command surgical instruments with a high precision to a desired position, we propose a switching servoing scheme. First, the instrument is driven to a Cartesian coordinate (reconstructed using stereopsis of the 3D endoscope) which is in sight of the camera, employing position-based servoing. Since the 3D reconstruction suffers from a certain error (caused by the mentioned intrinsic system errors) we subsequently switch to image-based servoing to overcome the remaining distance.

Given a target pose or position that the robot is to reach, visual servoing aims to minimize an error $e(t)$, typically defined by

$$e(t) = s(m(t), a) - s^* \tag{13}$$

where s^* represents the target pose, $s(m(t), a)$ the measured pose, $m(t)$ the measured image feature points and a any potential additional knowledge about the system. This includes calibration information of the camera as well as intrinsic parameters of 3D models of objects. The function $s(m(t), a)$ characterizes the end point of the tool tip of an instrument carried by the robot. In

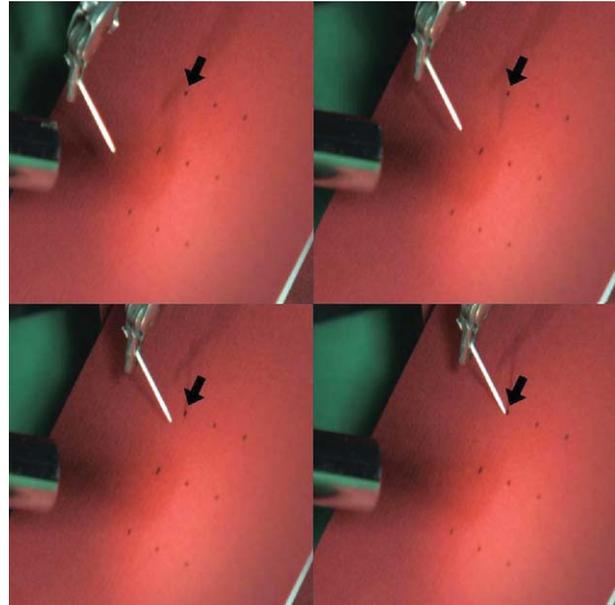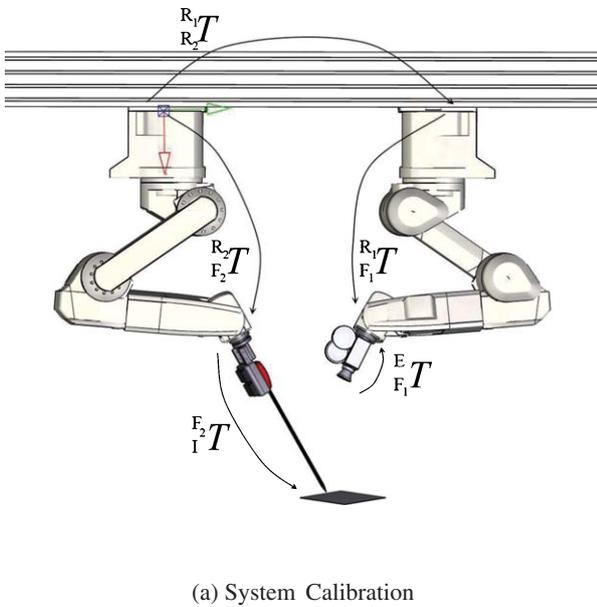(a) System Calibration   (b) Visual Servoing

Fig. 6. Illustration (a) exemplifies the intrinsic system aberrations that yield to a comparable bad overall system accuracy, i.e. the error of the flexible carbon fiber shaft of the instruments cannot be calibrated perfectly. The top left image in (b) shows the initial position of a needle that is to be driven to the marked dot. The final posture is reached in the bottom right picture.

PBVS the position of the tracked features is extracted from the camera image coordinates and projected to the world frame by the mapping a, determined during camera calibration. The target position can be extracted from image features in a similar way. While PBVS minimizes the error $e(t)$ in world coordinates and the camera is treated as a 3D positioning sensor, IBVS directly tries to find a mapping from the error function to a commanded robot motion.

3.2.1. Instrument control

As mentioned above, PBVS is used to drive the instrument to a reconstructed point which is located within the view of the camera. As soon as this point is reached, the remaining distance to the target goal is minimized in image coordinates. In many IBVS scenarios the camera is directly attached to the robot which is to be commanded (eye-in-hand configuration) and therewith the velocity of the camera ξ is calculated. In our setup, the instrument and the endoscope is carried by two different robots and the calculated velocity ξ has to be transformed to the robot that carries the instrument.

A single image feature, for instance the tip of an instrument or a carried needle is tracked in both left and right camera coordinates. The feature vector $s = (x_L, x_R)^T = (u_L, v_L, u_R, v_R)^T$ comprises these coordinates:

$$s(t) = \begin{bmatrix} u(t) \\ v(t) \end{bmatrix} \tag{14}$$

Its derivative $\dot{s}(t)$ is referred to as image feature velocity. It is linearly related to the camera velocity $\xi = [\upsilon\ \omega]^T$, which is composed of linear velocity υ and angular velocity ω. The relationship between the time variation of the feature vector s and the velocity in Cartesian coordinates ξ is then established by

$$\dot{s} = L_s\xi \tag{15}$$

where L is the *interaction matrix* or *image Jacobian* [5]. The interaction matrix L_x related to an image point $x = (u, v)^T$ reads as follows:

$$L_x = \begin{bmatrix} -\frac{1}{z} & 0 & \frac{u}{z} & uv & -(1+u^2) & v \\ 0 & -\frac{1}{z} & \frac{v}{z} & 1+v^2 & -uv & -u \end{bmatrix} \tag{16}$$

Variable z represents the depth of a point relative to the camera frame. There exist different ways to approximate the value of z, for example via triangulation in a stereo setup or via pose estimation. Most of the

existing methods assume an calibrated camera, even the quality of the calibration has few effects on the achieved precision. Some approaches even assume a constant depth of the tracked features. Though a stereo camera system is on hand, it is not used for depth estimation. In fact, variable z is estimated via the kinematic chain of the system. Using this method, L can be updated on-line and the method can easily be transfered to miscellaneous camera system. For instance, we equipped another robot arm with a second monocular Fujinon™ endoscope that provides a different viewing angle for the scene. Using equations (13) and (15) we obtain $\dot{e} = L_e\xi$ and our final control law

$$\xi = \lambda L_e^+ e \tag{17}$$

where λ is a positive gain factor and L_e^+ the Moore-Penrose pseudo-inverse of L_e.

As mentioned above, a single visual feature s is tracked in the left and right image. Equation (15) is then rewritten as

$$\begin{bmatrix} \dot{x}_L \\ \dot{x}_R \end{bmatrix} = \underbrace{\begin{bmatrix} L_L \\ L_R\ {}_L^R V \end{bmatrix}}_{=L_s} \xi_L \tag{18}$$

The spatial motion transform V to transform velocities expressed in the right camera frame R to the left camera frame L is given by ${}_L^R V = \begin{bmatrix} {}_L^R R\ S(t){}_L^R R, & 0\ {}_L^R R \end{bmatrix}^T \cdot \xi_L$, where $S(t)$ is the skew symmetric matrix associated with the linear translation vector t and where (R, t) is the transform from the left to the right camera frame. During the alignment with a desired target point I five degrees of freedom have to be controlled: the linear velocities $\upsilon_x, \upsilon_y, \upsilon_z$ are derived from the visual servoing process and the angular velocities ω_x, ω_y are determined by the trocar constraint (movements perpendicular to the incision direction are prohibited at the fulcrum). The calculated velocities correspond to the camera robot, but have to be transformed to the robot that is holding the surgical tool as depicted in Fig. 6(a).

The precision of the proposed control scheme was verified in several experiments, whereas the image-based part and the position-based servoing were considered separately. To measure the positioning error, the instrument was holding a needle in its jaws that had to be aligned with the target. Different target points were marked on a countertop and placed freely within the working space.

With regards to position-based servoing the absolute distance between the desired and the reached position was measured. The found aberration was (as expected) not satisfying and ranged between 3 and 7 mm, depending on the location in the working area (and also in the image, since the camera suffers from radial distortions). The image-based part was verified in a similar way, with the instruments being visible for the camera at the beginning of the procedure. Chosen images from the positioning process are depicted in Fig. 6(b). The tracking of the pinpoint was performed with a frame-by-frame color segmentation in HSV space. The achieved error did not exceed 1.3 mm and was evenly distributed over the entire image, since image-based servoing is marginally susceptible to camera parameters. The method was applied to automated tissue piercing with circular needles [30]. Therefore, a laser-pointer was mounted on one of the instruments and used as pointing device. The projected light dot indicates the favored incision point of the needle on the tissue and serves as desired position for the visual servoing. As the point is reached, the instrument is commanded in a way that the pinpoint of the needle performs a circular movement and pierces the tissue. In combination with the presented knot-tying autonomous suturing could be made possible.

4. Force prediction based time-delay compensation in surgical telepresence systems for incision processes

The transfer of surgical skills from a human operator to a technical system cannot be achieved if the unrealis-tic force-display is presented. It arbitrates the interpretation of the executed tasks and shall lead to the introduction of the increasing failure rate of the selective automation. One important variable which cause a distortion to the force feedback perception is time-delay. Given that time-delay is referred as a characteristic of network infrastructures, the delay cannot be altered physically [18]. The synchronization of master and slave can be lost and causes instabilities in the control loop for large time-delays. For diminutive time-delays, a low-pass filter may be applied and stabilize the control loop by eliminating high-frequency movements of the operator and of feedback forces. However, the application of low-pass filters results in an unrealistic perception which renders objects softer than they are in reality [26]. For surgical telepresence, this matter may produce fatal errors to tissue since operators make their movement decisions exceedingly relying on information fed back by the haptic device. Considering this problem, the control scheme is extended by integrating a force prediction algorithm [6, 8] as depicted in Fig. 7(a). The algorithm predicts the state or event occurring between the tool of the teleoperator and its remote environment in real time. The simulated perception of the tool position and the contact force are constantly fed back to the operator and enables a compensation of time-delays by eliminating the need of a delayed measured contact force from the actual force sensor at the end-effector of the teleoperator. For the interaction between a scalpel and a soft-body (such as tissue), a contact model is required to represent geometrical deformation as well as to precisely render the incision force.

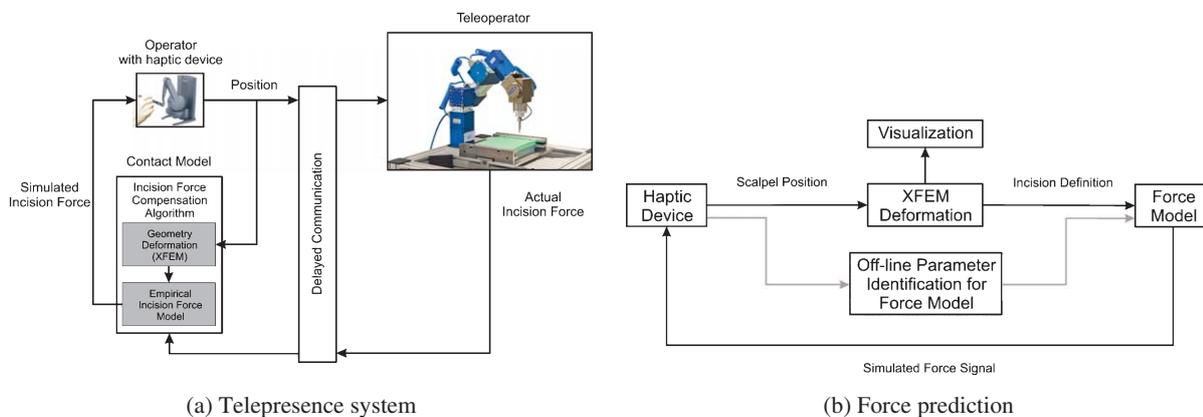(a) Telepresence system   (b) Force prediction

Fig. 7. (a) Depicts the telepresence system with time-delay compensation using real-time force prediction. (b) shows the diagram for the force prediction algorithm utilizing XFEM simulation and the incision force model.
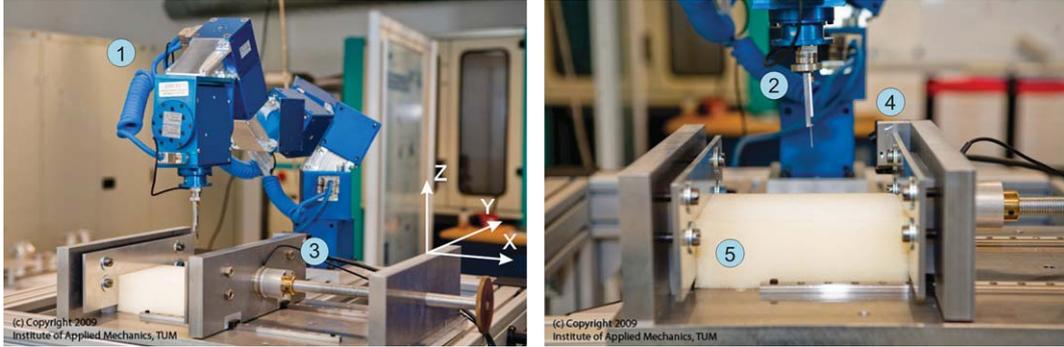
Fig. 8. Empirical modeling of the incision force: (1) 6 DoF modular robot, (2) force sensor & scalpel: Measuring incision forces, (3) force-torque sensor: Measuring tension forces, (4) laser distance sensor: Measuring the total displacement, (5) test object.

To efficiently represent constant geometry changes, we utilize a real-time linear XFEM (*eXtended Finite Element Method*, cp. 4.1.2.). XFEM is well known for its computing efficiency compared to the conventional FEM with remeshing technique [10, 23]. The haptic rendering of the incision force is done employing the incision force model referred in One et al. [23] (cp. 4.1.2.). The proposed incision force prediction algorithm is depicted as a flow-diagram in Fig. 7(b).

4.1. Modeling of force model for haptic rendering in incision process

We divide the incision mechanics in a soft-body with a scalpel into two stages. In first stage, the scalpel pierces the surface of the soft-body and forms a cut. Before the incision is made, the force can be calculated as the nodal external contact force f_{ext} using XFEM (Equation (21)). In second stage, as the scalpel has entered the surface, the incision force is composed of the material resistance during piercing and the friction between scalpel and the soft-body, hence the incision force $f_{scalpel}$ is employed instead (Equation (23)). The key element to the precise rendering of the incision force model is the correct identification of the material parameters (Equation (23)).

For this purpose, the entire incision process is performed with the aid of a robot (cp. Fig. 9). It is programmed to cut a test object by slowly pierce the scalpel into the object ("first stage"). Thereafter the robot draws the scalpel to form a cut in the test object ("second stage"). The incision forces are measured by a force sensor (Fig. 8, no. 2), mounted at the robot end-effector. Deformations of the test object can be

constantly monitored via a laser distance sensor (no. 4), while the second force sensor measures (no. 3) the axial tension force. Silicon is chosen for its elastic similarity to represent an artificial skin (no. 5). The parameters of unknown material properties are determined empirically in offline experiments.

4.1.1. Dynamic deformation in incision process with XFEM

To simulate the dynamic geometry deformation during the incision process, the material discontinuities caused by the cut have to be modeled efficiently. The classical FEM approach remeshes all elements of the body in order to defy the edge of the incision and rebuilds the system matrices for the next simulation step according to the actual geometry after the cut [22]. Unfortunately, this procedure generates new DoFs and enlarges the system matrices. In general, conventional FEM with remeshing technique requires more computational expense as the system matrices becomes larger. In contrast, the XFEM approach does not alter the initial system matrices, but enriches the cut element locally [2]. The computational expense is dramatically reduced, compared to conventional FEM [10].

To calculate the deformation, XFEM approximates a static displacement in the incised body $u(x)$ as a result of the interpolation of the standard finite element part and the additional global discontinuous enrichment function multiplied by a shape function with the locally added DoF a_j (cp. 19). The shape functions for the added DoF can be assumed to be identical to the standard ones, hence $\Phi_j^{*k} = \Phi_i(x)$. Index $i = 1, \ldots, n_{node}$ corresponds to the number of standard nodes in an element and $j = 1, ..., n_{node}$ refers to the number of enriched nodes. Variables $k = 1, ..., n_{cut}$ represent the

(a) Vertical incision force

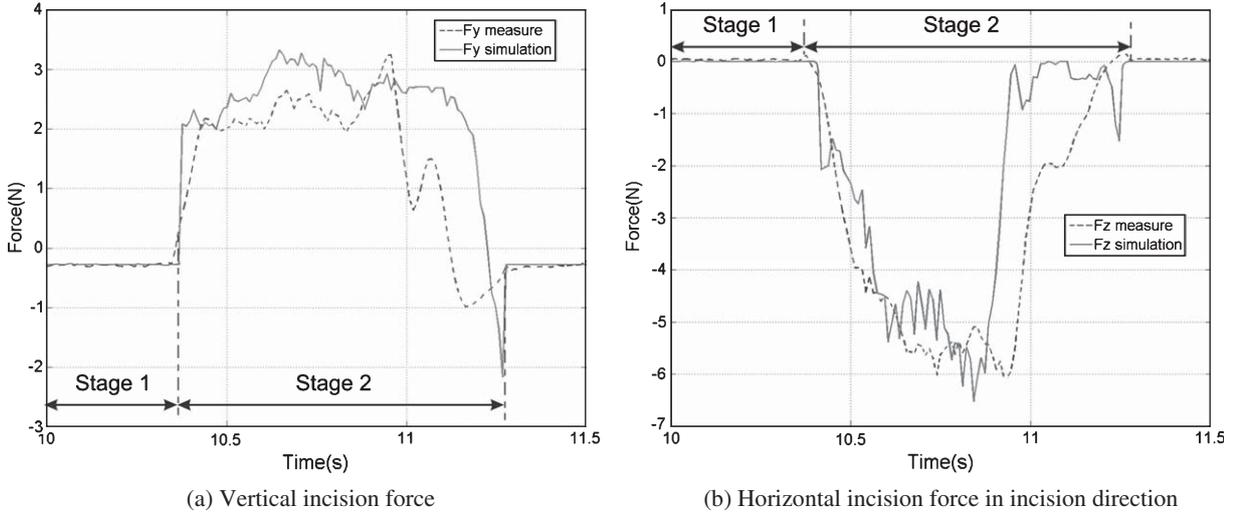(b) Horizontal incision force in incision direction

Fig. 9. Precision of force model versus the actual measurement.

set of discontinuities in the element (the number of cuts found in the element). The discontinuity function is chosen to avoid the complication regarding the treatment of boundary conditions. Hence, shifted enrichment functions are used, which are zero at all nodes [10].

$$u(x) = \sum_{i=1}^{n_{node}} \Phi_i(x)u_i + \sum_{k=1}^{n_{cut}} \sum_{j=1}^{n_{node}} \Phi_j^{*k}(x)\Psi_j^k(x)\, a_j^k \quad (19)$$

$$f_{ext_i} = \sum_{j=1}^{n_{node}} K_{ij}^{UU} u_i + \sum_{j=1}^{n_{node}} K_{ij}^{UA} a_j \qquad (20)$$

$$f_{ext} = M^{enr}\ddot{u} + (\alpha M^{enr} + \beta K^{enr})\dot{u} + K^{enr}u, \quad (21)$$

and

$$K^{enr} = \begin{bmatrix} K^{UU} & K^{UA} \\ K^{AU} & K^{AA} \end{bmatrix},\ K_{ij}^{UU} = K_{ij},$$

$$K_{ij}^{UA} = \left(\frac{V_a}{V}\Psi_{aj} + \frac{Vb}{V}\Psi_{bj} \right) K_{ij},$$

$$(22)$$

$$K_{ij}^{AU} = \left(\frac{V_a}{V}\Psi_{ai} + \frac{Vb}{V}\Psi_{bi} \right) K_{ij},$$

$$K_{ij}^{AA} = \left(\frac{V_a}{V}\Psi_{ai}\Psi_{aj} + \frac{Vb}{V}\Psi_{bi}\Psi_{bj} \right) K_{ij}$$

The term f_{ext_i} in Equation (20) is the nodal external contact force that acts on a node and is not identical to the incision force. Variable f_{ext_i} denotes the contact force that causes the deformation of a soft-body

after incision. The force f_{ext_i} is used in the incision process during the first stage. The formulation of the dynamic deformation equation of XFEM is a direct approach of a conventional FEM [1]. The mass M^{enr}, damping D^{enr} and stiffness K^{enr} matrices are formulate with the enrichment terms included in Equation (21). As mentioned earlier, the original system matrix from the conventional FEM will not be altered, hence K_{ij}^{UU} will be equal to K_{ij}. In [10], the original stiffness matrix K_{ij} is extended by the enrichment stiffness terms. Therefore, K^{enr} (cp. Equation (22)) is referred, whereas indices a and b represent two divided volumes in the incised element. The value of the enrichment stiffness terms (K_{ij}^{UA}, K_{ij}^{AU} and K_{ij}^{AA}) depends on how the element is cut which accords to the divided volume (V_a, V_b) and the volume of the element before the cut (V). The same approach is also applied to the mass matrix M^{enr}. The damping matrix relies on the Reyleigh formula.

4.1.2. Incision force model

As proposed earlier, the prediction of the incision during "second two" of the process cannot be realized using Equation (20) or Equation (21). Equation (23) refers to the second stage of the incision mechanics based on [23]. The variable ϵ is interpreted as the material stiffness constraint that represents the Hookean spring behavior of the resisting force between the scalpel and the cut object. Variable τ denotes the strength of the viscous friction-force and mg is the

weight of the scalpel. Variable r determines the position of the scalpel, where r_0 refers to the starting position of the incision.

$$f_{scalpel} = \epsilon(r - r_0) + \tau\frac{\partial r}{\partial t}(r - r_0) + mg \qquad (23)$$

4.2. Evaluation of incision force prediction algorithm

The precision of the proposed incision force modeling method is presented in Fig. 9. The algorithm effectively and automatically reproduces the incision force throughout the process. The experiments do not only help to identify the unknown material parameters in the model, but also demonstrate the essential force difference in both stages during the incision process. Since the scalpel can be assumed to be sharp, it will pass though the test object immediately after the first contact with barely noticeable changes in the incision force (transition between "first" and "second" stages). This can be interpreted as error or as a loss of the force feedback signal. The output of the algorithm is sufficient enough to compensate delayed force feedback signals in the real surgical teleoperation over-delayed network.

5. Conclusions and outlook

We presented a skill transfer framework in the context of minimally invasive robot-assisted surgery. The presented approach is based on a scaffolding framework. The basic idea of scaffolding is to utilize the superior knowledge of the trainer about the task to provide a framework of hints in order to assist the trainee. We have realized situated learning and scaffolding by means of a skill transfer architecture, which is based on abstract descriptions of tasks. Those are provided by the user and become an intrinsic part of the knowledge base of the system. The actual skill transfer is carried out via learning by demonstration. User demonstrations are decomposed into meaningful primitives with a finite automation, which are constructed from the definitions in the scaffolding framework.

In order to further improve the success rate of task instantiation we proposed the use of visually guided instruments that can overcome intrinsic system errors. First experiments have successfully been performed and showed an improvement in accuracy. Nevertheless these experiments are "point-and-pick" tasks, whereas

a desired position has to be defined once and is to be reached. Admittedly, an autonomous generation of visually followed paths is desirable and under ongoing work. Also the image processing and tracking part is under improvement and extension (see [29] for first results). In a next step, we will combine the visual servoing with the thread detection of our system [21] in order to handle surgical suture material automatically during knot-tying. To handle limp suture material that changes its geometrical shape in an unpredictable way and to include it into a task description need further context information. How this can be done (e.g. by visual cues) is, however, a completely open question. Furthermore, interesting links exist between our scaffolding approach and the developments in the field of surgical process modeling and surgical "language" [24]. 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.

The optimal transfer of a surgical skill depends strongly on the correct perception of the operator. A synchronized and precise perception between the visual display and force feedback is required for a transparent surgical teleopertion. Time-delay in communication channels is one of the certain interferences which postpone the force feedback signal to arrive at the operator and therefore causes an unrealistic telesurgery experience. To overcome this problem, we introduced an incision force prediction algorithm based-on XFEM to compensate the time-delays and to provide a correct force perception. In doing so, XFEM was used to simulate the geometry deformation of an object during the incision process. In combination with the empirical parameter identification, the incision force prediction algorithm correctly reproduced the contact forces that act on the scalpel. The promising results from the incision force prediction algorithm encourages us to integrate it into the control loop of our system in the future.
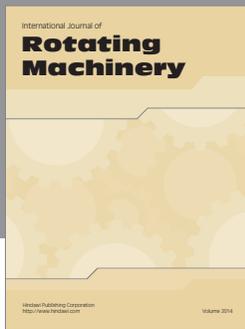
References

[1] K.J. Bathe, Efficient algorithms for layer assignment problems, *Finite-Elemente-Methoden*, Freeman, San Francisco, CA, 1973, pp. 63–83.

[2] S. Bordas, A. Legay and A. Gravouil, X-FEM mini-course, Department of Civil Engineering, Technical report, University of Glasgow, Lecture note from EPSFL short-course, 2007.

[3] E. Braun, C. Hasselbeck, H. Mayer, A. Knoll, S. Wildhirt, R. Lange and R. Bauernschmitt, Assessment of surgical experience for telemanipulated heart surgery, in *Proceedings of the World Congress on Medical Physics and Biomedical Engineering [CD-ROM]*, Seoul, Korea, August 2006.

[4] C. Cao, C. MacKenzie and S. Payandeh, Task and motion analyses in endoscopic surgery, In *Proceedings ASME Dynamic Systems and Control Division*, Atlanta, USA, 1996, pp. 583–590.

[5] F. Chaumette and S. Hutchinson, Visual servo control, basic approaches, *Robotics & Automation Magazine, IEEE* **13**(4) (2006), 82–90.

[6] S. Clarke, G. Schillhuber, M.F. Zaeh and H. Ulbrich, The effect of simulated inertia and force prediction on delayed telepresence, MIT Press, *PRESENCE* **16**(5) (2007), 543–558. Special Topic: Teleoperators and Virtual Environments.

[7] B. Espiau, Effect of camera calibration errors on visual servoing in robotics. In *The 3rd International Symposium on Experimental Robotics*, Springer-Verlag, London, UK, 1994, pp. 182–192.

[8] A. Golle, H. Ulbrich and F. Pfeiffer, Real-time simulation of non-smooth contacts in telepresence, In *Industrial Technology, 2003 IEEE International Conference on, 10–12 December 2003.* 2, pp. 675–679, 2003.

[9] P. Hynes, G. Dodds and A. Wilkinson, Uncalibrated visual-servoing of a dual-arm robot for MIS suturing, In *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006, The First IEEE/RAS-EMBS International Conference on Feb.* 2006, pp. 420–425, 2006.

[10] L. Jeřábková, *Interactive Cutting of Finite Elements based Deformable Objects in Virtual Environment*, Department of Informatic, published Ph.D. dissertation, RWTH Aachen, Germany, 2007.

[11] H. Kang, *Robotic Assisted Suturing in Minimally Invasive Surgery*, Rensselaer Polytechnic Institute, Troy, New York, 2002.

[12] M. Kitagawa, A. Okamura, B. Bethea, V. Gott and W. Baumgartner, 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*, T. Dohi and R. Kikinis, eds, Lecture Notes in Computer Science; vol. 2488, Berlin/Heidelberg, Springer-Verlag, September 2002, pp. 155–162,

[13] A.R. Lanfranco, E.A. Castellanos, J.P. Desai and W.C. Meyers, robotic surgery: A current perspective, *Annals of Surgery* **239**(1) (2004), 14–21.

[14] M.J. Mataric, Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics, MIT Press, 2000, pp. 391–422.

[15] H. Mayer, I. Nagy and A. Knoll, Kinematics and modeling of a system for robotic surgery, In *On Advances in Robot Kinematics*, J. Lenarcic and C. Galletti, eds, Kluwer Academic Publishers, Sestri Levante, Italy, June 2004, pp. 181–190.

[16] H. Mayer, I. Nagy, A. Knoll, E. Braun, R. Lange and R. Bauernschmitt, Adaptive control for human-robot skill-transfer: Trajectory planning based on fluid dynamics, In *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, April 2007, pp. 1800–1807.

[17] H. Mayer, I. Nagy, A. Knoll, E. Schirmbeck and R. Bauernschmitt, The Endo[PA]R system for minimally invasive robotic surgery, In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, September 2004, pp. 3637–3642.

[18] M. Mitsuishi, Y. Iizuka, H. Watanabe, H. Hashizume and K. Fujiwara, Remote operation of a micro-surgical system, **2** (1998), 1013–1019.

[19] M. Mitsuishi, S. Tomisaki, T. Yoshidome, H. Hashizume and K. Fujiwara, Tele-micro-surgery system with intelligent user interface, In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 2, 2000, pp. 1607–1614.

[20] J.A. Morgan, J.C. Peacock, T. Kohmoto, M.J. Garrido, B.M. Schanzer, A.R. Kherani, D.W. Vigilance, F.H. Cheema, S. Kaplan, C.R. Smith, et al., Robotic techniques improve quality of life in patients undergoing atrial septal defect repair, *The Annals of Thoracic Surgery* **77**(4) (2004), 1328–1333.

[21] I. Nagy, *3D Situs Reconstruction in Minimally Invasive Surgery*, Technische Universität München, Munich, Germany, 2008.

[22] H.W. Nienhuys, *Cutting in Deformable Objects*, published Ph.D. dissertation, Institude for Information and Computing Sciences, Department of Informatic, Utrecht University, 2003.

[23] K. Ono, G. Schillhuber and H. Ulbrich, XFEM approach to real-time incision haptic feedback for surgical simulation, In *International Conference on Extended Finite Element Method – Recent Developments and Applications*, Aachen, Germany, 29–30 September 2009.

[24] C.E. Reiley, H.C. Lin, B. Varadarajan, S. Khudanpur, D.D. Yuh and G.D. Hager, Automatic recognition of surgical motions using statistical modeling for capturing variability, In *Medicine Meets Virtual Reality*, vol. 132, 2008, pp. 396–401.

[25] S. Schaal, J. Peters, J. Nakanishi and A. Ijspeert, Learning movement primitives, In *Springer Tracts in Advanced Robotics: International Symposium on Robotics Research*, vol. 1805, Springer-Verlag, October 2004.

[26] G. Schillhuber, *Robotergestützte Modellidentifikation und Simulation von deformierbaren Körpern*, published Ph.D. dissertation, Department of Mechanical Engineering, Technische Universität München, 2009.

[27] K. Shoemake, Animating rotation with quaternion curves, In *Proceedings of the International Conference on Computer Graphics and Interactive Techniques*, San Francisco, CA, July 1985, pp. 245–254.

[28] A.C. Smith and K. Hashtrudi-Zaad, Smith predictor type control architectures for time delayed teleoperation, *International Journal of Robotics Research* **25**(8) (2006), 797–818.

[29] C. Staub, C. Lenz, G. Panin, A. Knoll and R. Bauernschmitt, Contour-based surgical instrument tracking supported by kinematic prediction, In *Proceedings of the IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2010.

[30] C. Staub, T. Osa, A. Knoll and R. Bauernschmitt, Automation of tissue piercing using circular needles and vision guidance for computer aided laparoscopic surgery, In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2010.

[31] R. Taylor and D. Stoianovici, Medical robotics in computer-integrated surgery, *Robotics and Automation, IEEE Transactions on* **19**(5) (2003), 765–781.

[32] S. Umeyama, Least-squares estimation of transformation parameters between two point patterns, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(4) (1991), 373–380.

[33] H. Wakamatsu, A. Tsumaya, E. Arai and S. Hirai, Manipulation planning for knotting/unknotting and tightly tying of deformable linear objects, In *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2005, pp. 2516–2521.