ToolBot: Robotically Reproducing Handicraft

Kim Wölfel, Jörg Müller, Dominik Henrich

ToolBot: Robotically Reproducing Handicraft

Kim Wölfel[1][0000−0003−3274−3346], Jörg Müller[2][0000−0002−4971−9126], and Dominik Henrich[1][0000−0003−0250−2728]

[1] Chair for Robotics and Embedded Systems, University of Bayreuth, 95447 Bayreuth, Germany
{kim.woelfel,dominik.henrich}@uni-bayreuth.de
[2] Chair of Serious Games, University of Bayreuth, 95447 Bayreuth, Germany
joerg.mueller@uni-bayreuth.de

Abstract. We present ToolBot, a robotic system for creating handicraft with hand-held tools. Through using our process users are able to work with their favorite tools when producing handicraft. We observe the production process using motion capture. We provide a trajectory editor, which allows non-robotics-experts to semi-automatically edit the tool motion. Finally, ToolBot executes the movements using a lightweight robot arm. This is non-trivial for three reasons. First, the original tool trajectory might be non-replicable due the robots limited workspace, collisions, or singularities. Second, the robots maximum velocities and accelerations can be exceeded. Third, the user might want to let the robot use a different tool or replicate downscaled tool motion. We resolve these issues, such that the hand-held tool can be mounted on a robot arm, which can then mass-produce the handicraft. We evaluate the usability of ToolBot through a user study.

Keywords: Fabrication; Human-Robot Interaction; Robot Programming by Demonstration.

1 Introduction

Artisanal handicraft is arguably the first personal fabrication technology. Carrying a long tradition, an enormous variety of hand-held tools and techniques exist for creating objects. A large fraction of the world's population possess considerable skill in operating these tools. Handicraft with hand-held tools allows for expressivity and detail. The main limitation of handicraft is that it does not scale for mass production. Produced objects are unique, and the effort of production grows linearly with the number of objects produced.

Industrial technologies allow a production of objects on a large scale. A huge variety of production technologies, including milling, injection molding, forming etc. have been developed. This greatly reduced the cost of producing additional objects, once an object is designed and a production line is set up.

Recently, personal fabrication machines, such as 3D printers, have been available at low cost. Additive, subtractive, and formative manufacturing allows in-

dividuals to produce, and mass-produce, self made objects. There has been extensive research in how to enable individuals to easily produce objects using these technologies. New approaches add considerable power and flexibility to the design process compared to handicraft. However, due to the different manufacturing process used, objects fabricated with mass-fabrication technologies do have very different appearances and detail from handicraft. Each fabrication technology comes with its unique fabrication artifacts and constraints.

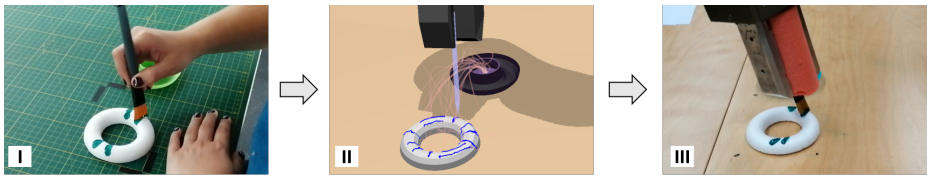

Fig. 1. Using ToolBot, users can produce handicraft with their personal hand-held tools. Tool motion is recorded using a Motion Capture System (I). Using our trajectory editor, non-robotics experts can then fix errors semi-automatically and adjust the tool movement to robot capabilites or other aspects and validate the execution with a simulation (II). The real-world robot can then reproduce the handicraft arbitrarily often, possibly at home or in a central factory (III).

We propose to solve this problem with the seemingly simple idea of replicating the production process using the hand-held tool and robotic technology (see Figure 1). Users handcraft their objects working with their own favorite tools. The tool movement is captured using motion capture technology. We provide a trajectory editor that allows non-robotics experts to semi-automatically fix errors and adjust the tool trajectory to robot capabilities. The tool is then installed on a robotic arm, which recreates the production process with a new object. In this way, the desired number of copies of the hand-crafted object can be created at low cost.

This approach poses three major technical challenges. First, the tool trajectory might be non-replicable for the robot, due to its limited workspace, collisions, or singularities. In this paper, we present several solutions for these problems consisting of a mix of automatic and manual modifications of the recorded trajectory. Second, our data shows that hand-held tools often reach velocities and accelerations which are beyond robot capabilities. We present an approach that deals with this problem by using a special interpolation technique. Third, the user might want to let the robot use a different tool or replicate downscaled tool motion. Our approach enables the use of different or scaled tools for modelling and replicating a motion.

One particular strength of our approach is that it works for almost any hand-held tool. This includes classical tools such as knifes and brushes, but also extends to modern tools such as Dremel and 3Doodler.

2 Handicraft and Mass Fabrication

Two major differences between handicraft and mass fabrication are that, first, they use very different tools and fabrication techniques, and second, that Handicraft is unique, while mass fabricated items are usually perceived as not being unique. Regarding the first point, each fabrication technology comes with its own unique fabrication artifacts and constraints. Injection moulding, for example, creates visible grates where the mould sides connect, where the material enters the mould, and at the ejector pins. Furthermore, products containing "overhangs" need to be produced from multiple parts. Materials that can be injection moulded are also very limited.

Hand-carved items, on the other hand, have very visible traces of the carving tool. The traces are often irregular, and of the pattern that is created by guiding the tool with a human hand. The entire workpiece often contains irregularities which create the distinct "Handicraft" appearance. Materials are limited to those soft enough to be carved with human strength, such as wood. Many people prefer the appearance of Handicraft objects (e.g., hand carved) to those created by mass-production (e.g., injection moulded).

Regarding the second point, hand crafted items are often perceived as unique, while mass-produced items are usually not. The impact of the reproducability of art on its perception has been discussed by Walter Benjamin [3]. He argues that some art has been reproducible since ancient times, e.g., terracotta and bronze. Modern technologies like printing and photography extended the range of art that could be reproduced, thereby changing the nature of art and becoming a new art themselves. Benjamin coins the term of the "Aura" of a unique artpiece, describing the effects of its unique existence at one location. The "Aura" encompasses chemical and physical changes over time, as well as the history of owners and events happening to the art piece. Only in this case it makes sense to speak of the "original" and of authenticity. We think that it is important to realize that ToolBot is only one new reproduction technology in a long history of these. One difference is that ToolBot replicates the production process itself, not merely the end product. This means for example, that the influence of the raw material is preserved in the final product. The economic impact of such a technology on artisans is also not forseeable. On one hand, the ability to create handicraft without additional human labor might increase competition for them. On the other hand, the ability to mass-produce their work could allow artisans to reach more customers and compete with industrial companies. In this paper, we concentrate on the technical aspects of this reproduction process. While we want to mention the possible implications of our work to media and art, we prefer to leave these aspects to future work in the fields that possess more expertise in these areas, such as media studies.

3 Related Work

3.1 Personal Fabrication

Personal fabrication has been described as "the ability to design and produce your own products, in your own home, with a machine that combines consumer electronics with industrial tools" [10]. In recent years, a wealth of knowledge and new ideas has been generated on the topic of personal fabrication. A comprehensive summary is provided by [2]. The majority of research deals with additive (e.g., 3D printing), followed by subtractive (e.g., milling, laser cutting) and formative fabrication technologies, i.e. by applying mechanical forces, restricting forms, heat, or steam to form a material into a desired shape [11]. Many researchers investigate how to design models for 3D printing or milling machines in an easier way, such as [20].

3.2 Assisted Handicraft

Several smart handheld tools have been developed, which are actuated jointly by the users' hands and computer-controlled actuation, e.g. [13] or [21]. An excellent overview of this topic is provided by [24]. Furthermore, systems allow users to create 3D models both by forming clay and by editing the model digitally, such as ReForm [19], or by wax coiling using a hand-held actuated extruder (*D-Coil* [15]). The *Makers' Marks* System [17] allows the fabrication of functional objects via annotation stickers and *Drill Sergeant* [18] applies the concept of software tutorials onto physical craft processes through augmented tools.

3.3 3D Photocopying

Several mechanisms have been proposed for "photocopying" existing objects. Scotty [12] is a "3D teleporter" that destructively scans an object, while reproducing it at a different location, as well as ensuring that only one copy exists. The work in [4] also maintains the physical deformation behavior of objects while photocopying. Even commercial 3D photocopying machines have been available for this purpose [22].

3.4 Painting Robots

When it comes to replicating paintings or decorating eggs some systems, like the WaterColorBot, AxiDraw or the EggBot, have been presented which are able to generate astonishing results. Most of these systems take an actual image as an input and then reproduce it on a sheet of paper or a 3D surface if it is possible to fixate it in the system. These approaches differ from our framework, because the input is limited to a 2D graphic. While this is satisfactory for painting an image using water color or projecting it on a 3D surface, task like modelling clay with an appropriate tool is hard or impossible to realize.

3.5 Robot Programming by Demonstration

Robot programming by demonstration [1] enables users to teach robots by performing motions without having to write code. Many approaches perform the teaching phase by tracking the human wrist [16] or guiding the robot arm by hand (*kinesthetic programming*) [14], [23]. The later approaches work well in pick-and-place scenarios, but may limit the users in cases where more dynamic motions are needed. We found only two papers mentioning direct tracking of the users' tool [8], [6].

The contribution presented in [8] is a novel tracking system with a custom active marker and a stereo camera. The marker can be attached to a tool and the recorded motion can then be executed by a robot arm. However, the proposed tracking system is considerably less accurate than state-of-the-art motion capture systems. The application proposed is that of tracking a large spray paint gun. Motions are slow due to the large mass of the gun, and the required accuracy is relatively low. The user is required to record the motion at the actual robot location within the workspace of the robot. Therefore, the authors of [8] are able to directly map their recordings to robot motion without modification.

In [6] an approach is presented which records not only the position-controlled human motion but also takes care of the occuring forces. This is done by utilizing a reciprocating electric carving tool. These tools are normally used to facilitate carving actions by lowering the force needed to penetrate the material. These forces can also be extracted and therefore used for later applications. The worker is tracked while performing a carving task a few times and the recorded motion is then used to train a neural network. The neural network is able to generate robot trajectories containing the performed carving motion for work pieces with varying grain direction, length or depth. Therefore, their system is able to perform learned carving *skills* through a multi-shot learning pipe-line.

Our contribution includes features of these two approaches to allow the replication of a complete manipulation (like [8]) together with a high accuracy ($<$ 1mm) and the possibility to perform the motion by oneself [6]. Additionally, we enable the user to edit the recorded data manually in a post processing step.

4 Methods

In this section we briefly describe important definitions from the area of robotics which are necessary to understand the rest of the paper. Desired poses of the robot arm (joint configurations) can be set in *joint space*, i.e. angles for each joint J_i of the robot arm (see Figure 2 right), or in *Cartesian space*, i.e. in the form of a transformation matrix in an *NSA* coordinate system (see Figure 2 left) for the tool center point (TCP) located at the end of the robot arm with a potential offset (ignoring multiple solutions in the case of redundant robot arms). Here, $\vec{n} \in \mathbb{R}^3$ is the direction perpendicular to the finger movement of an attached gripper, $\vec{s} \in \mathbb{R}^3$ is the slide direction of its fingers and $\vec{a} \in \mathbb{R}^3$ is the approach direction of the gripper. The transformation between the two

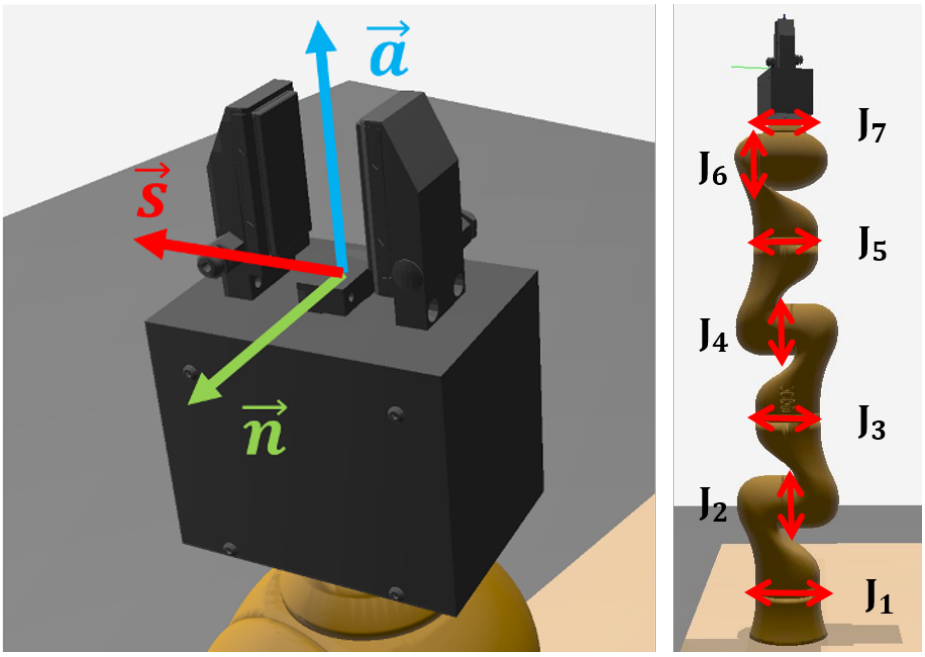

Fig. 2. NSA coordinate system of the gripper (left) and robot arm (right) with joints $J_i, i \in \{1, 7\}$ and orientations.

spaces is achieved by *forward kinematics* (FK), from joint to cartesian space, and *inverse kinematics* (IK), from cartesian to joint space. When moving the robot arm through given joint configurations, an adequate interpolation strategy is required.

While we want to imitate human motions, time-optimal trajectories between these configurations are important, so as to be able to replicate movement in a realistic way. During the execution, the necessary robot configurations are then computed using these trajectories.

5 Implementation

Our ToolBot system consists of five main steps: recording a tool's motion, transforming the tracking data into configurations in the robot workspace, editing the transformed trajectory by hand, validating the configurations using a robot simulation and executing the final motion on a real robot arm.

For recording the tool's motion, we built a motion capture setup comprising twelve OptiTrack Flex 3 cameras mounted on a cube-like rack (see Figure 3). This setup is located on the table the user is working at and can easily be adapted, if different camera poses are needed. We record and export the tracking data with

the *Motive* software. The tracked data is modified and transformed into a series of robot configurations using the analytical solution for a 6 Degrees of Freedom (DoF) robot arm presented in [7]. This is done automatically by a Python script we wrote using the libraries *pandas*, *numpy* and *scipy*. The trajectory editor we implemented (see Figure 4), which can be used to edit parts of the trajectory, as well as the robot simulator, which allows a virtual validation of the current robot motion, are implemented in C++. The robot simulator not only allows the execution of the robot motion in a virtual environment, but also renders contact points between the tool's tip and the workpiece (see Figure 5), which highlights the impact of editing the trajectory. The last step, the execution of the transformed data, is performed on a KUKA LWR IV robot arm with 7 DoF and a Schunk PG70 gripper (see Figure 2 right).

6 The ToolBot System

The limits of replicating hand-held tool movement by using a robot arm are set by the robot's maximum joint angles, velocities, accelerations, and jerks. This leads to limits in the Cartesian space (e.g. collisions or unreachable positions) and joint space (e.g. unreachable configurations). We adapt the recorded human movement to these limits via a data processing pipeline consisting of three major stages: data preprocessing, data editing, and data validation (see Figure 6).

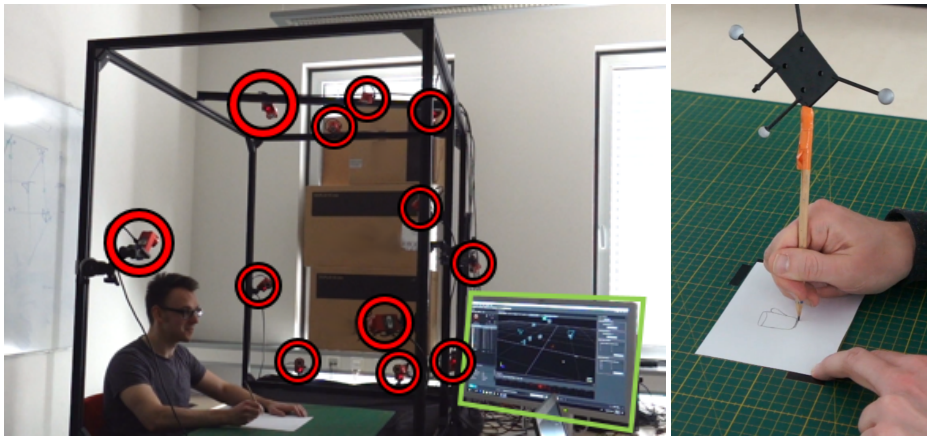

Fig. 3. Left: Optitrack Setup: Twelve Optitrack Flex 3 cameras (red circles) located around the executed motion and the visualization of the data using the Motive software (green rectangle) Right: Pencil with attached marker.

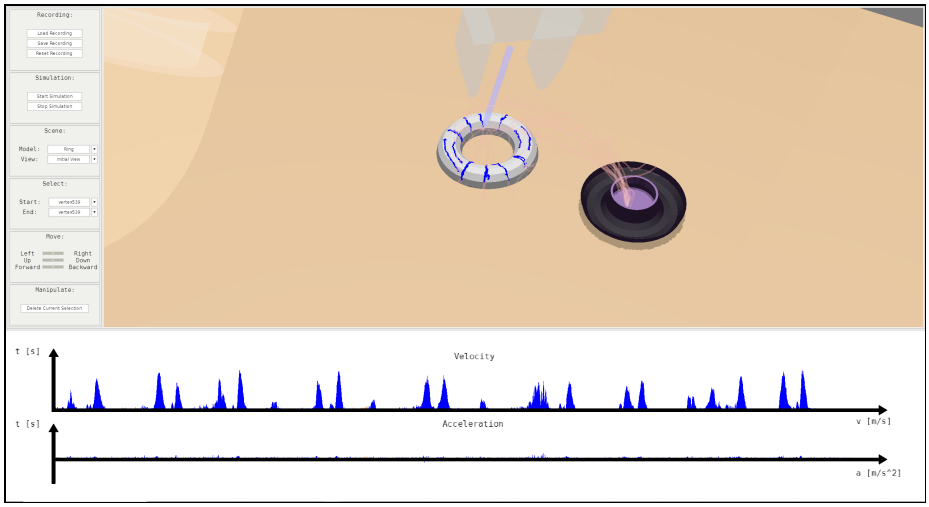

Fig. 4. The ToolBot Editor. Enables users to transform and delete selected vertices of a recorded motion (blue) in the rendering window (right) via mouse input or through GUI widgets (left). It also renders the velocity and acceleration data of the TCP in the lower left area.

6.1 Data Preprocessing

The main challenge during this stage is to modify the data exported from the motion capture software, such that irregularities and data loss are fixed as effectively as possible.

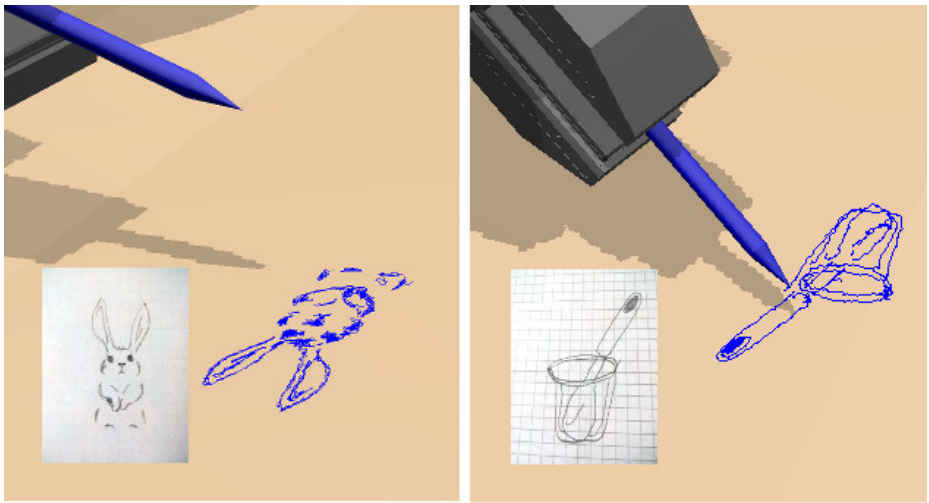

Fig. 5. Two example painting motions replicated in the robot simulation. Contact points are connected using blue lines.

Data Extraction Before a motion is recorded, we define the tool used as a rigid body with the Motive software, using several markers attached to the tool and a single marker placed inside of the workspace. The single marker is necessary to be able to define a rigid body, with a marker located at the tool tip. We then compute the tool-tip's pose by using the position of the single marker inside of the rigid body and the rigid body's orientation estimated by the motion tracking software. These poses are then transformed into a trajectory T_{init}, which stores the tool-tip pose for each time-step. This trajectory may be incomplete due to occlusions of the markers by the user or obstacles in the scene. We fix this data loss by a linear interpolation between valid entries.

If the original motion was performed on a non-rigid surface or a lot of noise is present in the recorded data, it may happen that some contact points are ignored, or the workpiece could even be penetrated. In this case, we perform another preprocessing step which estimates the transformation of the manipulated object and transforms the tracking data to fix surface irregularities. For example, in a drawing application we use the *Random Sample Consensus* (RANSAC) [9] to fit a plane into the drawing pane and project all tracking points lying behind this plane on its surface. At this point, a trajectory T_{prep} is available that stores the tool-tip position along with the corresponding tool orientation for the entire tracked motion.

Data Transformation The main challenge of this stage is to find an appropriate transformation of the tool tip trajectory T_{prep} into the robot workspace and an adequate fallback method to adjust the gripper's transformation, if there are still unreachable configurations after the transformation of T_{prep}.

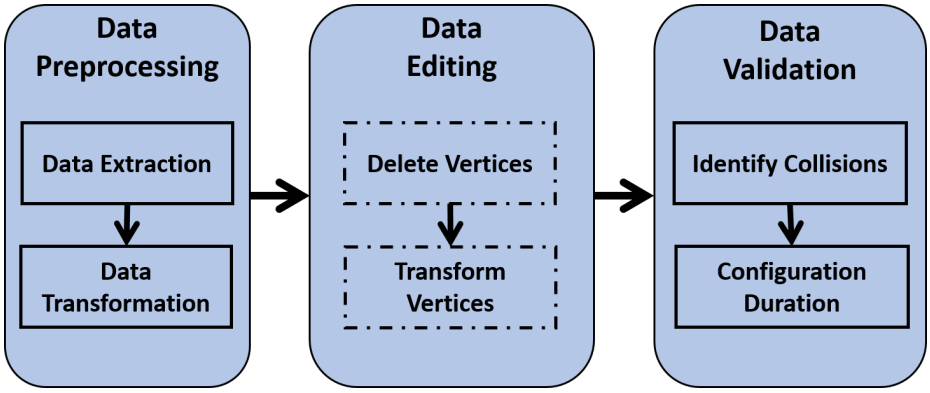

Fig. 6. ToolBot Data Processing Pipeline. Boxes with round corners represent main stages, boxes with sharp corners represent mandatory steps (straight lines) and optional steps (dashed lines).

We transform T_{prep} from the coordinate system in which it was recorded, into the robot's workspace as follows: Let $\vec{z}_{\text{wp}} \in \mathbb{R}^3$ be the normalized axis of the workpiece, pointing away from the underlying surface. We compute a homogenous transformation matrix $M_{\text{robot}} \in \mathbb{R}^{4\times4}$, which represents a rotation from $\vec{z}_{\text{wp}}$ onto the z-Axis of the robot base $\vec{z}_{\text{robot}} \in \mathbb{R}^3$ and a translation of the motion into the center of the robot's workspace. Therefore, the resulting trajectory T_{trafo} is computed for n tracking points as follows:

$$T_{\text{trafo},i} = M_{\text{robot}} \cdot T_{\text{prep},i}, \forall i \in \{0, \ldots, n-1\}. \tag{1}$$

All entries of T_{trafo} are then converted into a trajectory of NSA transformations T_{TCP} for the robot arm. This is done by projecting the tool-tip along the z-axis of its orientation for an appropriate length l_m. The TCP orientation is computed by transforming the rigid body's orientation. For example, in the case of a pen, the tip position $\vec{x}_t \in \mathbb{R}^3$ and orientation $\vec{q}_m$ are extracted from the tracking data. The rotation $\vec{q}_m$ is then used to rotate a unit vector $\vec{x}_z$ pointing into the z-direction with the length l_m to compute the TCP position via:

$$\vec{x}_{\text{TCP}} = \vec{x}_t + l_m \cdot \vec{q}_m \vec{x}_z. \tag{2}$$

These transformations are then converted into joint configurations using an IK algorithm. In our case, we set the robot's joint J_3 (see Figure 2, right) to zero, and use an analytical IK solution to overcome problems like singularities associated with numerical solutions for 7 DoF robot arms. This results in a list of valid joint configurations for the motion, as long as these exist.

Instead of immediately marking an execution as not executable if an unreachable transformation is found, we give the user the opportunity to adapt the trajectory by hand in the Data Editing stage. At this point we mark the configuration as not reachable with the user's configuration and attach a default rotation for the gripper. In the future, we plan to investigate more elaborate fallback methods using motion planning approaches.

6.2 Data Editing

In case users are not satisfied with the result of their work, they would normally have to repeat the work from scratch, which can be time consuming and tends to annoy the artist. To overcome this problem, we implemented a trajectory editor, which enables the user to manipulate the recorded trajectory. Similar to proportional editing of vertices in the 3D modelling software "Blender", users can manipulate single or sets of vertices by translating them along fixed directions (see Figure 7 (III) and (IV)) or deleting them (see Figure 7 (I) and (II)). At the moment the translation is done by mapping the users mouse input to a 3D motion. Depending on the currently locked 3D axis the vertices are then translated and highlighted as soon as invalid robot configurations are reached.

To avoid discontinuities in the trajectory, translating vertices does not only affect the selected vertices but also some of the adjacent vertices. Starting from the selected vertex, adjacent vertices are also manipulated, if the distance to

their preceding vertex does exceed a pre-defined distance d_{limit}. This guarantees, that the robot stays inside of its velocity limits while executing the trajectory. To preserve the smoothness of the trajectory while translating vertices, adjacent vertices are only manipulated by a fraction the distance or orientation displacement per time step. In our editor this is done using a weight function $\omega(d) \in [0, 1[$ computing the displacements depending on the distance $d \in \mathbb{R}$ to the selected vertex. While $\omega(d)$ should be symmetric around 0 and smooth, we use a wide normal distribution which results in a smooth transition between the vertices (see Figure 7).

To improve the usage of our editor for experts as well as non-experts, we also implemented three features which prevent the user from generating potentially invalid configurations. The first feature is, that the robot is rendered in

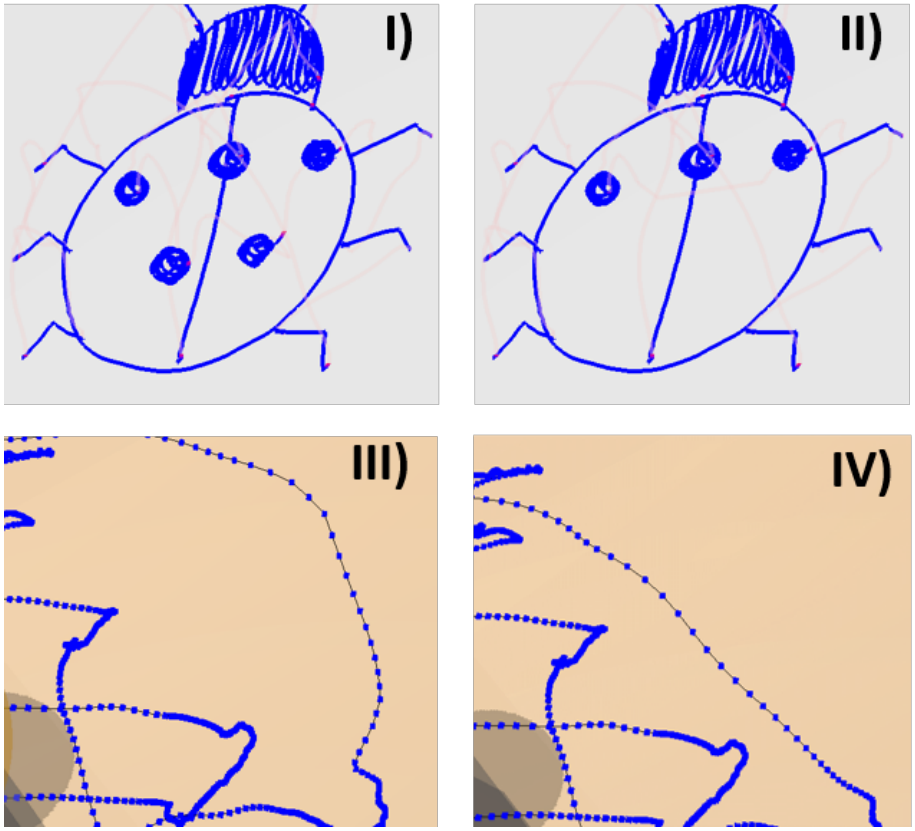

Fig. 7. Applications of our trajectory editor: Easily deleting parts from an original recording (I) while preserving the rest of it (II). Transforming vertices of the original trajectory (III) to correct unwanted movements (IV).

a transparent way while a vertex is selected. This is useful to check for potential collisions between the gripper and the environment. The second feature is highlighting non-reachable configurations in red. As long as these configurations are present in the trajectory, an execution is not possible. The last feature are the velocity and acceleration profiles rendered at the bottom of the editor. They show the velocities and accelerations necessary to move between vertices and can therefore be used to estimate to which extent a vertex can be manipulated. Those may be not of interest for every user of the system, but is helpful to estimate, if dynamic limitations are reached soon. These features lead to a trajectory which can be executed by the robot arm and satisfies the user's requirements.

6.3 Data Validation

The main challenge at this stage consists of two tasks. The first is to identify and handle collisions between the robot and its environment by adjusting T_{TCP} based on the type of collision. The second challenge is to find durations between two configurations that lead to the fastest possible replication.

Collision Detection Before the trajectory can be executed by a robot arm, the list of configurations is validated as follows: We first simulate the motion in our robot simulation framework, as described in the Implementation section, to automatically identify the location of collisions between the robot gripper and the table which may occur as a result of the difference in size between the gripper and a human hand. This has to be done, because such collisions corrupt the replicated motion and can lead to severe damage to the workpiece, tool, or robot. If such a collision is identified, the affected vertex of our trajectory is highlighted in red and the trajectory is marked as not valid. To be able to execute the manipulation by the robot the user has to fix these vertices by manipulating the trajectory with our trajectory editor (see Figure 4).

Configuration Duration The last pipeline step is to speed up the robot's execution. We use linear functions with parabolic blends presented in [7] to interpolate between our configurations Θ_i and speed up the execution by varying the durations d_i which specify the time the robot should take to move from one configuration to another. As long as d_i is high enough, the necessary linear path segment between each two parabolic path segments can be generated. Because of this, we start with an initial duration of d_{init} which is valid in most cases. We then iterate over all configurations and check, whether or not they are valid. While a configuration is valid, we adapt its duration by decreasing d_i by a constant d_Δ. If a configuration is not valid, we adapt its duration by increasing d_{i} by d_Δ until a valid result is reached. This is repeated until all configuration durations are set to a valid value, depending on d_Δ resulting in the fastest possible execution.

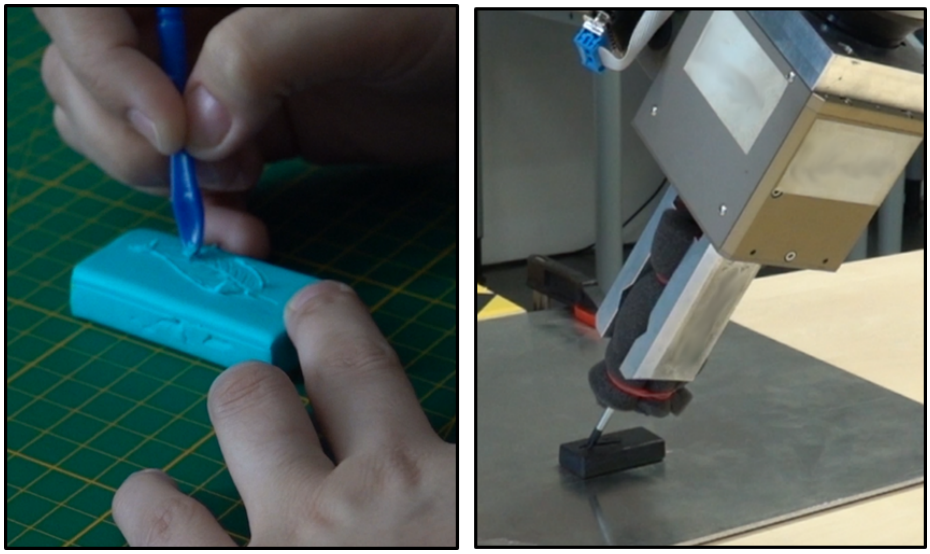

Fig. 8. Recording a motion with a Fimo modelling tool, length $l = 12.7$cm, (left) and its replication using a screwdriver, $l = 25.4$ cm, (right).

7 Scaling Tool or Workpiece

Besides the need to know the contact point computed in the data preparation stage, in later process steps, this also allows the replication of the same contact point trajectory using a different tool as long as the tool's geometry is known. We had already tested this feature several times with different tools (see Figure 8) and by scaling the tool. Until now, scaling the tool had to be done for each replication, because the gripper is usually larger than the tool used while recording the motion. By doing this, the user is even able to use an abstract tool for recording the motion, which is replaced by the desired tool in the real-world execution.

An advantage of using the transformation in the data transformation stage is that the recorded motions can be imitated not only for workpieces of the same size, but also for scaled work pieces. This enables users for example to model details on enlarged workpieces which are then executed on smaller workpieces by the robot.

8 User Study

In order to evaluate the usability and user experience with using Toolbot, we conducted a user study. We observed users while using the Toolbot system and conducted a semi-structured interview with them. As questionnaires, we used the System Usability Scale (SUS) and the User Experience Questionnaire (UEQ).

Fig. 9. Examples from the user study. The top row shows five drawing examples (top: participant, bottom: robot arm). The bottom row shows five modelling examples (left: participant, right: robot arm). Some artifacts, like the missing part of the duck, originated from calibration problems, others, like the additional lines in the most left and most right image, from the trajectory editing during the study.

Additionally, we asked whether they experienced the manipulated trajectory and the object produced by the robot arm as their own work.

Overall 15 users (5 female, 10 male) with a mean age of 27 participated at our study. Before they began they had to rate their experience regarding the three topics programming, robotics and 3D editors on a scale of 1 to 7, and name any 3D editors they had already used. Their mean experience in programming was 4.1, the mean experience in robotics was 2.9 and the mean experience in using 3D editors was 3. 50% of the users had been working with a 3D editor before.

8.1 Procedure

Users created two pieces with an optically tracked tool. They drew on a postcard and carved an image into modelling paste (see Figure 9, row 1 and 2). The system transformed the recorded data as described in the data preprocessing section 6. Users then edited their trajectories in our Toolbot editor. During the editor use, we used a think aloud protocol. After deleting and transforming data points, the users did run the robot motion in the simulation and saved their results. The users then watched the real robot arm executing their edited motion. For each motion the user was able to attach the tool used to the gripper and examine the robot motion. After the users have saw the motion, they filled the SUS, UEQ and the custom questionnaire. Finally, we performed a semi-structured interview.

8.2 Results

The results produced by the robot arm are shown in Figure 9. Overall the robot was able to replicate the recorded and edited motions in most cases. Some prob-

lems were caused by calibration problems, e.g. the correct positioning and differences in the height of the modelling clay or the problems in the mounting of the tool used. Most of the users felt excited about seeing the robot replicating their work.

We evaluated the SUS following the rules from [5] and scored 68.4, so we can conclude that users rated the usability of our system as marginally above average. Although 50% of the participants had no experience with 3D editors 66% of them rated our system as easy to use and that various functions were well integrated. One main complaint was the mouse control for navigating the camera in the virtual world, which resulted in user statements like: "The view is zooming too fast.", "The other 3D editors i am using have a different control."

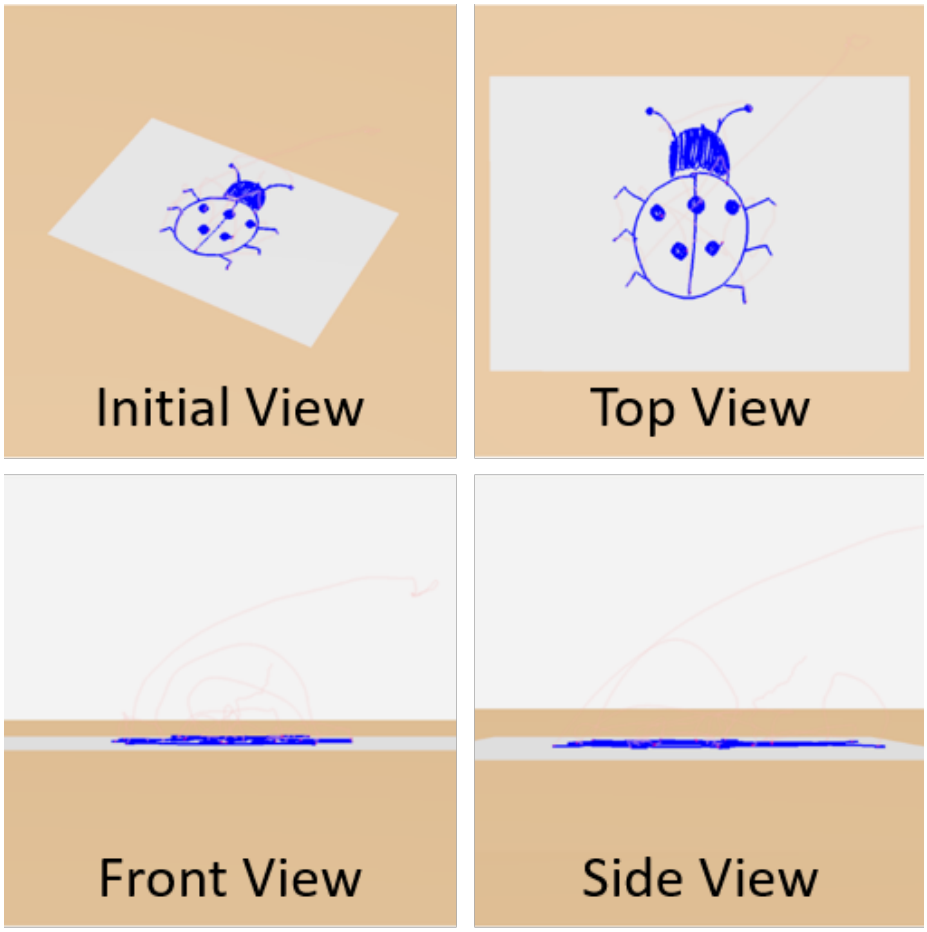

Fig. 10. Selectable camera views in our ToolBot editor.

and "I would like to have pre defined camera views". This inspired us to add a dropdown menu which enables the user to switch to pre defined camera views (see Figure 10) and updated the camera control. Another comment of some users was: "Now i select the line on the work piece.", which did not lead to success because it was only possible to select the TCP trajectory above the line. We fixed this problem by transforming the TCP trajectory into a tooltip trajectory and highlighted contact points in solid blue, free space points in transparent red and selected points in solid orange (see Figure 11). It was also mentioned by the users that operating the system was easier in the second task.

We evaluated the UEQ with the tool presented on the UEQ website [3]. Results for the UEQ are shown in Figure 12. The values show that users generally enjoyed the experience of using the system. Lower scores for efficiency and dependability show that, due to the prototype nature of Toolbot, users were not yet able to solve all tasks without effort.

In the custom questionnaire and semi-structured interview, 73% of the users stated that they did not feel restricted by the system when creating their handicraft. Some participants stated that there were some restrictions by holding the tool in a steep enough angle such that the gripper would not collide with the table. We observed that, while working with the hand-held tools, contacts between the users hand and the table were common. Additionally, users often changed the grip of the tool to reach different poses. Because such contacts with the surface should be avoided for robots, and the robot cannot change the grip, this poses challenges for the robot motion. Our current solution is to extend the length of the tool for the robot. An alternative solution would be to place the

[3] https://www.ueq-online.org/

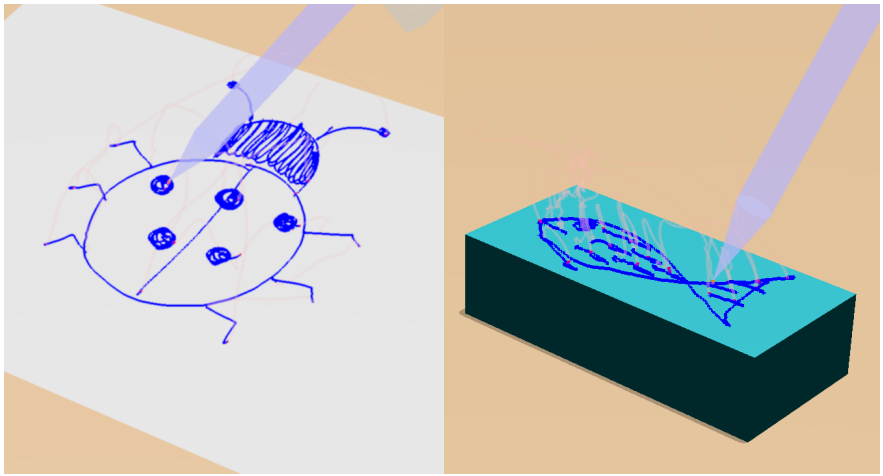

Fig. 11. Revised trajectory visualization (contact points in solid blue, free space points in transparent red) for a drawing and modelling trajectory.

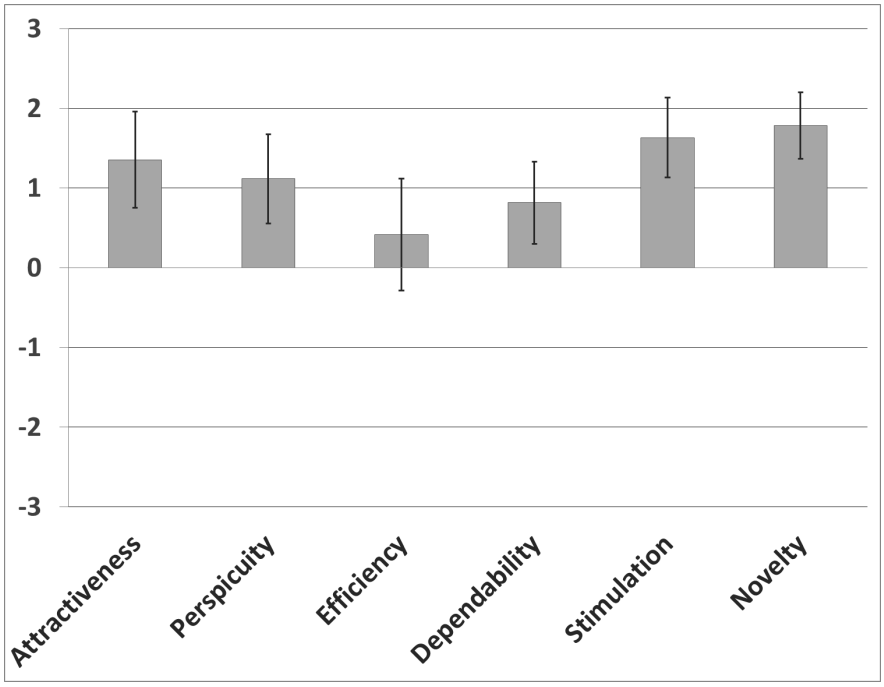

Fig. 12. Evaluation of the performed UEQ between -3 (bad) and 3 (good).

workpiece on an elevated platform, so it can be reached from flatter angles by the robot. After seeing the robot result, 80% of the users identified with the edited tool motion, because they manipulated them by themselves. Even 66% of the users still identified the result produced by the robot as their own work. The remaining users disagreed because of imperfections resulting from calibration errors, positioning of the work pieces and the insufficient accuracy of the robot arm used. A majority of the participants (80%) would give away the result as a present or sell it. Conclusively, the majority (73%) of the participants would use such a system in the future. The remaining users disagreed, because of the calibration errors and restrictions while recording the motion. These findings show that more work is to be done to further increase the accuracy of the robot replication. A major area of future work would be to detect variations in the size of the work piece, and develop accurate calibration methods of the work piece relative to the robot arm.

9 Discussion

In this section limitations of the presented system are discussed. At this point, tool motions can be recorded, are automatically transformed into robot configurations, and can be manipulated per configuration. Forces occuring at the

workpiece or tool are not considered yet, but plannend to be integrated in future work. As stated in section 6, unreachable configurations are reset to a default orientation at the moment as a fallback. We are planning to improve this fallback in the future by using further visual feedback or estimating more appropriate fallback orientations. Since the robotarm geometry differs from that of a human arm, adding a platform on which the workpiece is manipulated would be helpful in future work as well.

10 Conclusion

We have presented a novel approach to robotically creating handicraft with hand-held tools. ToolBot is directed towards non-robotics-experts and allows them to mass-produce their handicraft without writing code. We have presented a novel approach for transforming tool motions executed by humans to robot arms. Our approach allows the scaling of the tool used, automatically identifies collisions between the robot arm and the scene, and offers solutions that allow an execution of the recorded motion, even if this would not have been possible with the initial motion data. To ensure the validity of the final robot configurations, our framework contains a robot simulation which is able to perform the motion using a simulated robot controller. To enable the user to delete or adjust mistakes we implemented a trajectory editor for translating and deleting samples of the recorded trajectory.

The system is clearly still in prototype stage. To develop it towards a product, particularly the accurate calibration of the work piece relative to the robot needs to be improved. However, the result of the user study we conducted shows, that we can already conclude that the robotic replication of hand-held tool movements is a promising idea that warrants further investigation in the future. We hope that robotic fabrication with traditional tools might inspire and augment research in fabrication in Human-Computer Interaction, Robotics, and related disciplines.

Acknowledgements

This work has partly been supported by Deutsche Forschungsgemeinschaft (DFG) under grant agreement He2696-18.s

References

1. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and autonomous systems **57**(5), 469–483 (2009)
2. Baudisch, P., Mueller, S.: Personal fabrication: State of the art and future research. In: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. pp. 936–939. ACM (2016)
3. Benjamin, W.: The work of art in the age of its technological reproducibility, and other writings on media. Harvard University Press (2008)

4. Bickel, B., Bächer, M., Otaduy, M.A., Lee, H.R., Pfister, H., Gross, M., Matusik, W.: Design and fabrication of materials with desired deformation behavior. In: ACM Transactions on Graphics (TOG). vol. 29, p. 63. ACM (2010)
5. Brooke, J., et al.: Sus-a quick and dirty usability scale. Usability evaluation in industry (1996)
6. Brugnaro, G., Hanna, S.: Adaptive robotic training methods for subtractive manufacturing. In: Proceedings of the 37th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA). pp. 164–169. Acadia Publishing Company (2017)
7. Craig, J.J.: Introduction to robotics: mechanics and control, vol. 3. Pearson/Prentice Hall Upper Saddle River, NJ, USA (2005)
8. Ferreira, M., Costa, P., Rocha, L., Moreira, A.P.: Stereo-based real-time 6-dof work tool tracking for robot programing by demonstration. The International Journal of Advanced Manufacturing Technology **85**(1-4), 57–69 (2016)
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: Readings in computer vision, pp. 726–740. Elsevier (1987)
10. Gershenfeld, N.: Fab: the coming revolution on your desktop–from personal computers to personal fabrication. Basic Books (2008)
11. Kolarevic, B.: Digital fabrication: manufacturing architecture in the information age (2001)
12. Mueller, S., Fritzsche, M., Kossmann, J., Schneider, M., Striebel, J., Baudisch, P.: Scotty: Relocating physical objects across distances using destructive scanning, encryption, and 3d printing. In: Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction. pp. 233–240. ACM (2015)
13. Mueller, S., Lopes, P., Baudisch, P.: Interactive construction: interactive fabrication of functional mechanical devices. In: Proceedings of the 25th annual ACM symposium on User interface software and technology. pp. 599–606. ACM (2012)
14. Orendt, E.M., Fichtner, M., Henrich, D.: Robot programming by non-experts: intuitiveness and robustness of one-shot robot programming. In: Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on. pp. 192–199. IEEE (2016)
15. Peng, H., Zoran, A., Guimbretière, F.V.: D-coil: A hands-on approach to digital 3d models design. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. pp. 1807–1815. ACM (2015)
16. Rueckert, E., Lioutikov, R., Calandra, R., Schmidt, M., Beckerle, P., Peters, J.: Low-cost sensor glove with force feedback for learning from demonstrations using probabilistic trajectory representations. arXiv preprint arXiv:1510.03253 (2015)
17. Savage, V., Follmer, S., Li, J., Hartmann, B.: Makers' marks: Physical markup for designing and fabricating functional objects. In: Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology. pp. 103–108. ACM (2015)
18. Schoop, E., Nguyen, M., Lim, D., Savage, V., Follmer, S., Hartmann, B.: Drill sergeant: Supporting physical construction projects through an ecosystem of augmented tools. In: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. pp. 1607–1614. ACM (2016)
19. Weichel, C., Hardy, J., Alexander, J., Gellersen, H.: Reform: integrating physical and digital design through bidirectional fabrication. In: Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology. pp. 93–102. ACM (2015)

20. Weichel, C., Lau, M., Kim, D., Villar, N., Gellersen, H.W.: Mixfab: A mixed-reality environment for personal fabrication. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 3855–3864. CHI '14, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2556288.2557090, http://doi.acm.org/10.1145/2556288.2557090
21. Willis, K.D., Xu, C., Wu, K.J., Levin, G., Gross, M.D.: Interactive fabrication: new interfaces for digital fabrication. In: Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction. pp. 69–72. ACM (2011)
22. ZEUS: The world's first all-in-one 3d printer/copy machine. (2018), http://www.zeus.aiorobotics.com
23. Zoliner, R., Pardowitz, M., Knoop, S., Dillmann, R.: Towards cognitive robots: Building hierarchical task representations of manipulations from human demonstration. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. pp. 1535–1540. IEEE (2005)
24. Zoran, A., Shilkrot, R., Goyal, P., Maes, P., Paradiso, J.A.: The wise chisel: The rise of the smart handheld tool. IEEE Pervasive Computing **13**(3), 48–57 (2014)